

COLEGIO DE CIENCIAS Y HUMANIDADES
INSTITUTO DE INVESTIGACIONES EN MATEMATICAS APLICADAS Y EN SISTEMAS
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

S I M A S E T
SISTEMA DE MANEJO DE SERIES DE TIEMPO
T E S I S
que para optar el grado de
MAESTRO EN CIENCIAS DE LA COMPUTACION
presenta:
MARIO DANIEL HERDOIZA POSSO

MEXICO, D.F.

MARZO DE 1980



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

SIMASET

C O N T E N I D O

página

I.	INTRODUCCION	1
II.	DESCRIPCION GLOBAL DE SIMASET	
	1. ADMINISTRADOR DEL SISTEMA	5
	2. PROCEDIMIENTOS DE PROCESAMIENTO	6
	3. ARCHIVOS DE DATOS	7
III.	DESCRIPCION DETALLADA DE SIMASET	
	1. PROGRAMAS DE APOYO	14
	1.1 ARCHIVO1	14
	1.2 VALORES	15
	2. ADMINISTRADOR DEL SISTEMA	
	2.1 EL PROGRAMA SIMASET	15
	2.2 EL PROGRAMA PRIN/EJECUTA	15
	2.3 COMANDOS DE SIMASET	
	2.3.1 COMANDOS DE SERVICIO GENERAL	16
	2.3.2 COMANDOS DE PROCESAMIENTO NUMERICO DE LA INFORMACION	17
	2.3.3 DESCRIPCION DE LOS COMANDOS DE SERVICIO GENERAL	
	2.3.3.1 CREASERIE	18
	2.3.3.2 CREATABLA	19
	2.3.3.3 ESCRSERIE	19
	2.3.3.4 ESCRTABLA	21
	2.3.3.5 LEESERIE	22
	2.3.3.6 LEETABLA	23
	2.3.3.7 LISTSERIE	24

2.3.3.8	LISTTABLA	25
2.3.3.9	ACTUSERIE	25
2.3.3.10	ACTUTABLA	26
2.3.3.11	QITASERIE	27
2.3.3.12	QITATABLA	28
2.3.3.13	TERMINA	28
2.3.4	DESCRIPCION DE LOS COMANDOS DE PROCESAMIENTO DE LA INFORMACION	
2.3.4.1	SEN/1(MAX)	29
2.3.4.2	SEN/2(MIN)	29
2.3.4.3	SEN/3(MEDIA)	30
2.3.4.4	SEN/4(NORM2)	30
2.3.4.5	SEN/5(NORMINF)	31
2.3.4.6	SEN/6(NORM1)	31
2.3.4.7	SEN/7(DESUIA)	32
2.3.4.8	SES/1(SUMA)	32
2.3.4.9	SES/2(MULT)	33
2.3.4.10	SES/3(CONVOL)	34
2.3.4.11	TAT/1(SUMA)	35
2.3.4.12	TAT/2(PREDICT)	36
2.3.4.13	TAT/3(CONTROL)	37
2.3.4.14	TAN/1(MAX)	38
2.3.4.15	TAN/2(MIN)	39
2.3.4.16	TAN/3(MEDIA)	39
2.3.4.17	SS/1(FILTRA)	40
2.3.4.18	SS/2(FILTRA)	41
2.3.4.19	SS/3(FILTRA)	41

2.3.4.20	SS/1(ESPECTRO)	42
2.3.4.21	SS/2(ESPECTRO)	43
2.3.4.22	SS/3(ESPECTRO)	43
2.3.4.23	SS/1(SUAVIZA)	44
2.3.4.24	SS/2(SUAVIZA)	44
2.3.4.25	SS/3(SUAVIZA)	45
2.3.4.26	SS/(ESCALAR)	46
2.3.4.27	SET/1(IDENT)	47
2.3.4.28	SET/2(IDENT)	48
2.3.4.29	SET/3(IDENT)	48
2.3.4.30	SET/1(HISTO)	48
2.3.4.31	SET/2(HISTO)	48
2.3.4.32	SET/3(HISTO)	49
2.3.4.33	TAS/1(SIMULA)	49
2.3.4.34	TAS/2(SIMULA)	50
2.3.4.35	TAS/3(SIMULA)	50
2.3.4.36	GRA/1(CONT)	50
2.3.4.37	GRA/2(HISTO)	51
2.3.4.38	GRA/3(HISTO)	51
2.3.4.39	GRA/4(LINEAS)	51
2.3.4.40	GRA/5(PUNTOS)	52
3.	PROCEDIMIENTOS DE PROCESAMIENTO	52
4.	ARCHIVOS DE ALMACENAMIENTO	
4.1	ARCHIVOS DE SERIES	56
4.2	ARCHIVOS DE TABLAS	57
4.3	DESCRIPCION ESTANDARIZADA DE LOS ARCHIVOS	
4.3.1	VALORES	58

4.3.2 DIRUSER	59
4.3.3 ARCH1	60
4.3.4 DIRECTORIO	61
4.3.5 TABLA	62
4.3.6 DIRTAB	63
5. OPERACIONES PARA EL INGRESO DE NUEVOS COMANDOS A SIMASET	64
IV. FUNCIONAMIENTO Y RESULTADOS	
1. INGRESO AL SISTEMA	65
2. RESULTADOS Y EJEMPLOS	66
2.1 EJECUCION DE COMANDOS DE SERVICIO GENERAL	67
2.2 EJECUCION DE COMANDOS DE PROCESAMIENTO NUMERICO ...	74
2.3 EJECUCION DE COMANDOS DE GRAFICACION	87
V. CONCLUSIONES	99
BIBLIOGRAFIA	100
APENDICE A	
LISTADOS DE LOS PROGRAMAS	

I. INTRODUCCION .

Son comunes a los hombres y a las máquinas diversos procesos mecánicos y físico-sensoriales, pero todavía no se puede afirmar lo mismo respecto a los procesos intelectuales.

Casi todos los aspectos de la actividad humana varían y se transforman en el transcurso del tiempo. Entre otras, se pueden considerar a actividades económicas, agrícolas, técnicas, etc, las que también son influenciadas por modificaciones y transformaciones en el tiempo.

De ahí no es difícil entender las razones para estudiar ciertos modelos (de procesos "evolutivos") que se han llamado series de tiempo. [1]

La aplicación de estos modelos es en realidad más vasta, ya que el término "tiempo" se puede interpretar de una forma más general.

Así, "tiempo" puede referirse a cualquier parámetro lineal, como la medida de longitud de un objeto, su temperatura, su masa, etc. Entonces la llamada serie de tiempo consiste de valores de alguna otra variable tomados sobre el parámetro "tiempo".

Es clara la necesidad de un sistema que permita analizar series de tiempo.

El objetivo principal del presente trabajo, es el de desarrollar un sistema computacional, que maneje diversas funciones de almacenamiento y recuperación de la información, y de procesamiento de la misma, y que permita establecer un vínculo entre usuarios de series de tiempo y analistas de algoritmos para series de tiempo.

Considérese a una persona con un conjunto de datos de tipo numérico dependientes de un solo parámetro que previamente ha sido discretizado (series de tiempo). A esta persona le gustaría realizar una serie de operaciones sobre sus datos, tales que le permitan aprender de su estructura y seleccionar de acuerdo a ciertos criterios, diversos algoritmos de procesamiento numérico.

Para lograr esto, la persona debería :

- i) Diseñar y construir cómo y dónde van a estar sus archivos de almacenamiento de la información,
- ii) Implementar los procedimientos que desea que actúen sobre sus archivos, sean estos numéricos o de graficación,
- iii) Establecer formas de vincular a los archivos de almacenamiento

de la información, con los procedimientos de procesamiento de ésta, y
iv) Obtener resultados de las transformaciones que ha experimentado
la información.

Frente a este problema se desarrolla SIMASET como un sistema que
posibilita realizar los cuatro pasos indicados anteriormente.

II. DESCRIPCION GLOBAL DE SIMASET .

Uno de los métodos de descripción es el que analiza a los elementos o componentes de un todo pasando de lo general a lo particular.

SIMASET está constituido por tres componentes principales :

- procedimientos de procesamiento
- administrador del sistema, y
- archivos de datos

estos tres componentes están relacionados de tal manera que permiten establecer interacciones entre diversos usuarios y el sistema.

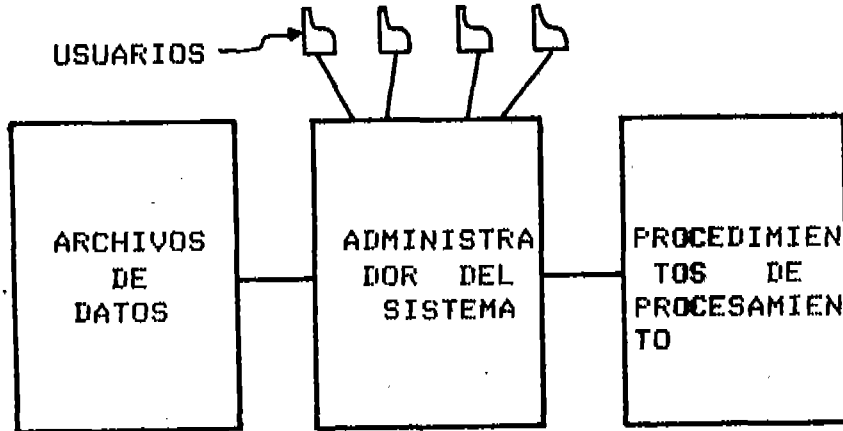


Figura 1.

Los procedimientos de procesamiento de la información están organizados en familias dependiendo del tipo de información que tienen como entrada, y de los resultados que producen. Las familias están diseñadas en forma de árboles que se relacionan con el administrador del sistema.

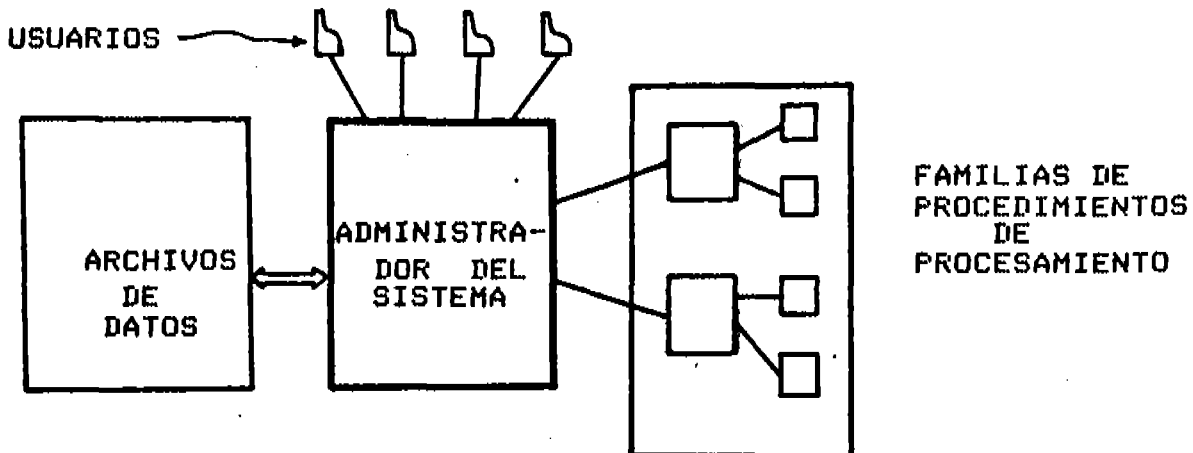


Figura 2.

El administrador del sistema está constituido por dos elementos, que son los que permiten establecer los mecanismos de relación entre:

- los usuarios y el sistema, y
- los procedimientos de procesamiento y los archivos de datos.

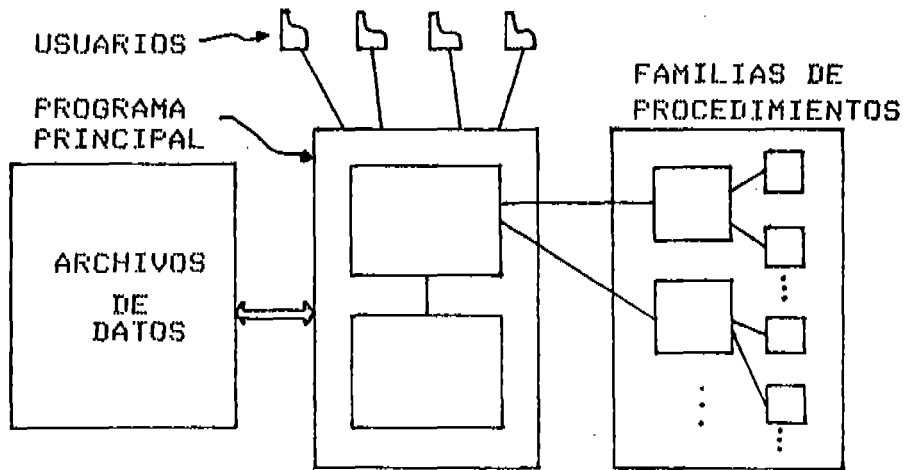


Figura 3.

Cada usuario puede generar sus propios archivos donde almacenaría la información numérica. Estos archivos están divididos en dos grupos diferentes :

- archivos que almacenan datos del tipo serie, y
- archivos que almacenan datos del tipo tabla.

DEFINICION.
SERIE

Se define como serie a un conjunto de datos que dependen de un parámetro lineal.

DEFINICION.
TABLA

Se define como tabla al polinomio:

$$A_0 Z^0 + A_1 Z^1 + A_2 Z^2 + \dots + A_N Z^N$$

donde los A_i , $i = 0, 1, 2, \dots, N$ son matrices de dimensiones $M \times P$. Una tabla es pues, un conjunto de $(N+1)$ matrices de dimensiones $M \times P$.

lo que se representa como un cubo.

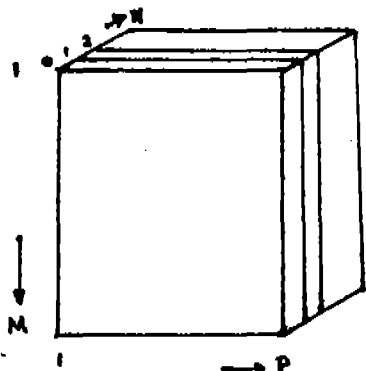


Figura 4.

Una vez dada una visión general del sistema, se procede a describir en una forma más detallada a los diferentes componentes del mismo.

II.1 ADMINISTRADOR DEL SISTEMA .

Como se mencionó antes, el administrador del sistema está constituido por dos elementos principales :

El primero es donde se incluyen ciertos procedimientos denominados de servicio general, siendo éstos los que permiten realizar operaciones de :

- generación de archivos,
- lectura/escritura de datos,
- modificaciones de archivos,

entre otras, el segundo es donde se llaman a todos los procedimientos que tiene el sistema. Estos procedimientos están relacionados directamente con comandos, a través de los cuales se establece una relación de los usuarios con los procedimientos que desean ejecutar.

El administrador del sistema está constituido por SIMASET que es el elemento a donde convergen todos los otros elementos y componentes del sistema, y PRIN/EJECUTA que es el elemento que sirve de mediador para establecer las relaciones entre los procedimientos de procesamiento y los datos.

Los procedimientos de servicio general son :

CREASERIE
CREATABLA

que se refieren a la generación de los archivos físicos para series y tablas respectivamente,

ESCRIBE
LEE

que permiten escribir y leer los datos en los archivos de almacenamiento,

LISTADO

que despliega o lista los datos numéricos que han sido almacenados,

ACTUALIZA QUITA

que realizan operaciones de actualización y de eliminación de los archivos de datos, y

TERMINA

que elimina toda la información que se refiere al usuario solicitante.

II.2 PROCEDIMIENTOS DE PROCESAMIENTO .

En este módulo existen dos tipos principales de procedimientos de procesamiento de la información :

- procedimientos de procesamiento numérico, y
- procedimientos de graficación

Los procedimientos de procesamiento numérico realizan operaciones estadísticas generales, de identificación, de determinación de espectros, de suavizamiento, de simulación, de filtrado, de predicción, etc.

Los procedimientos de graficación, realizan diversos tipos de gráficos sobre los conjuntos de datos almacenados.

Los procedimientos de procesamiento están organizados en familias de algoritmos, dependiendo de sus entradas y salidas. Estas familias en términos generales están conformadas por los siguientes grupos de algoritmos :

- i) SEN: Algoritmos que tienen como entrada una serie, y producen como salida un número.
- ii) SES: Algoritmos que tienen como entrada dos series, y producen como salida otra serie.
- iii) TAT: Algoritmos que tienen como entrada dos tablas, y producen como salida otra tabla.
- iv) TAN: Algoritmos que tienen como entrada una tabla y producen como salida un número.
- v) SS: Algoritmos que tienen como entrada una serie, y producen como salida otra serie.

vi) SET: Algoritmos que tienen como entrada una serie, y producen como salida una tabla.

vii) TAS: Algoritmos que tienen como entrada una tabla, y producen como salida una serie, y

viii) GRA: Por último se incluye como una de las familias de algoritmos de procesamiento a los algoritmos que tienen como entrada una serie y realizan diversos tipos de gráficos de sus elementos.

III.3 ARCHIVOS DE DATOS .

Un usuario puede generar hasta cuatro archivos relacionados con el almacenamiento de la información.

Cada archivo donde se almacena la información tiene relacionado un archivo auxiliar que sirve de directorio al archivo de almacenamiento.

Los nombres generales de los archivos son :

- {clave}/ARCHI/{id}
- {clave}/DIRECTORIO/{id}

que son los archivos de series, y

- {clave}/TABLA/{id}
- {clave}/DIRTAB/{id}

que son los archivos de tablas.

Con la descripción general y enumeración de los elementos que conforman el sistema, se incluye en el siguiente capítulo un diagrama de la estructura general del mismo.

III. DESCRIPCION DETALLADA DE SIMASET.

SIMASET (Sistema de Manejo de Series de Tiempo) está implementado en la B-6800 del Centro de Servicios de Cómputo de la Universidad Nacional Autónoma de México.

SIMASET utiliza diversos programas que permiten :

- almacenar
- actualizar, y
- procesar

la información numérica.

Existen tres clases principales de programas :(véase la Figura 5)

(i) Dos programas que tienen la función de generar archivos, y son : ARCHIVO1 que genera el archivo maestro DIRUSER que es el directorio de usuarios del sistema, y VALORES que almacena los comandos o instrucciones que puede ejecutar el sistema. A estos programas se los denomina "PROGRAMAS DE APOYO".

(ii) Los programas SIMASET y PRIN/EJECUTA. Simaset incluye a un conjunto de rutinas de servicio general, que posibilitan realizar acciones como:

- generación de archivos físicos (series, tablas)
- escritura
- lectura
- actualización
- eliminación

entre otras, y PRIN/EJECUTA realiza el llamado de todos los procedimientos del sistema. Estos programas conforman el ADMINISTRADOR DEL SISTEMA.

(iii) Programas que sirven para el procesamiento numérico de la información.

La estructura general de SIMASET, está desarrollada de tal manera que vincula a los archivos de datos almacenados en disco, con los diversos programas de procesamiento numérico.

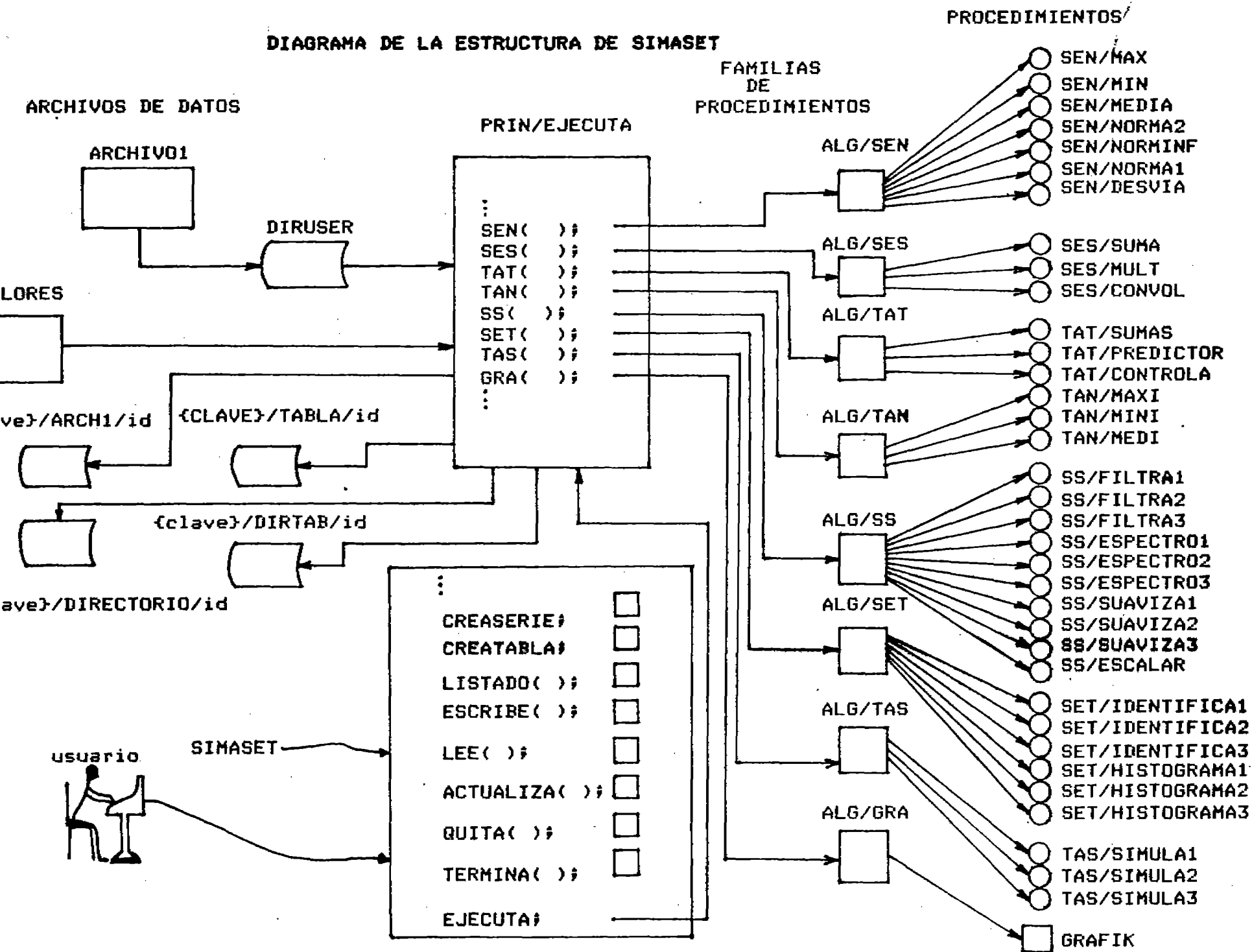
La idea principal de organizar a los algoritmos de procesamiento numérico en forma independiente, tiene su motivo en el hecho de facilitar la creación de una biblioteca de rutinas, que puedan ser usadas por SIMASET, o por algún otro tipo de sistema que se desarrolle.

Así se ve que SIMASET incluye ciertos módulos que son los que se denominan los padres de las familias de los procedimientos de procesamiento numérico. Estos módulos hacen uso de diversas rutinas de

servicio general que permiten entre otras actividades cargar a los conjuntos de datos desde disco, a memoria principal, para ahí transformarlos y nuevamente y siempre y cuando así sea requerido, almacenarlos en los dispositivos de memoria secundaria, en este caso disco.

Entre las familias de SIMASET, está la que grafica a los conjuntos de datos que se han almacenado, permitiendo obtener una apreciación visual de como se dan las transformaciones numéricas en ellos. Para tal efecto se hace uso del graficador Hewlett Packard del IIMAS.

DIAGRAMA DE LA ESTRUCTURA DE SIMASET



Uno de los problemas con un sistema como SIMASET, es el de agregar o modificar uno o más procedimientos de procesamiento. La dificultad es que en general esto requeriría la operación costosa de recompilar y religar todo el sistema.

Se analizan diferentes alternativas de como pudo haber sido desarrollado el sistema en respuesta a este problema.

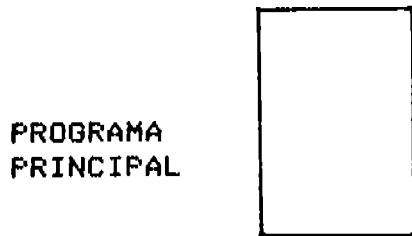


Figura 6.

La primera idea es de un sistema que consta de un sólo programa compilado (Figura 6).

Esta sufre de la obvia desventaja de que resulta costoso incluir nuevos módulos, o modificar los existentes, ya que se debe recompilar a todo el programa.

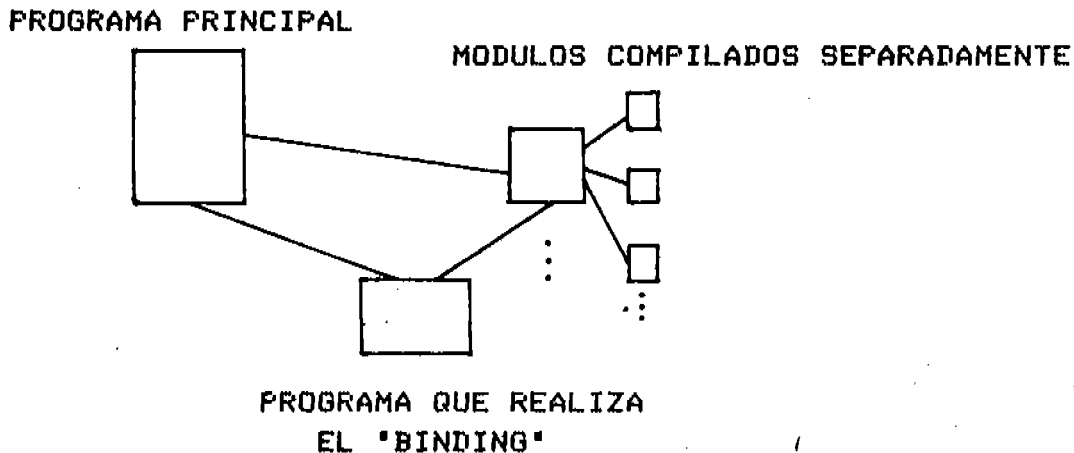


Figura 7.

Otra estrategia sería la de usar programas compilados separadamente que se ligan a través de un programa que realiza el "binding" de los diferentes componentes del sistema (Figura 7).

En este caso se facilita el hecho de incluir nuevos módulos, ya que sólo se deben compilar éstos separadamente y agregarlos en el

Programa del "bindings".

Este caso presenta la desventaja de la existencia tanto de archivos fuente como de archivos objeto, lo que significa que se tienen que abrir y cerrar todos los archivos con sus correspondientes buffers de "allocations" y de "deallocations".

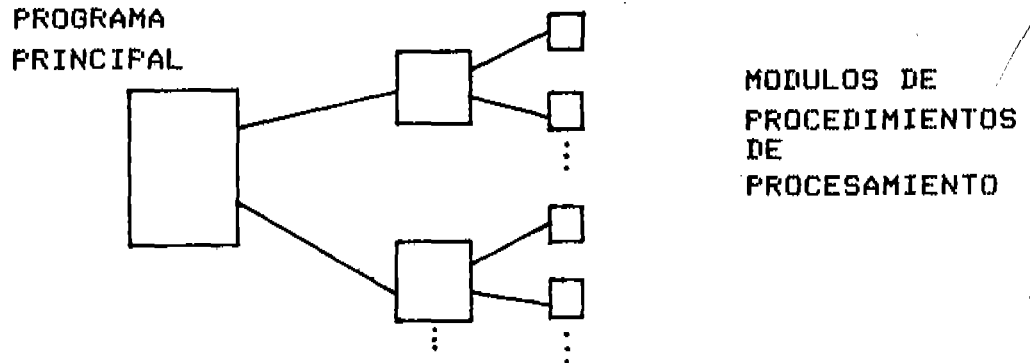


Figura 8.

El tercer diseño es de un sistema organizado por diversos archivos fuente, pero todos compilados y unidos en un sólo archivo objeto (Figura 8).

En este caso al incluir nuevos módulos, sólo se compilan éstos ligándolos directamente al programa principal.

Sin embargo existe el inconveniente de que si se accesa al programa principal, es necesario recompilar todo el sistema.

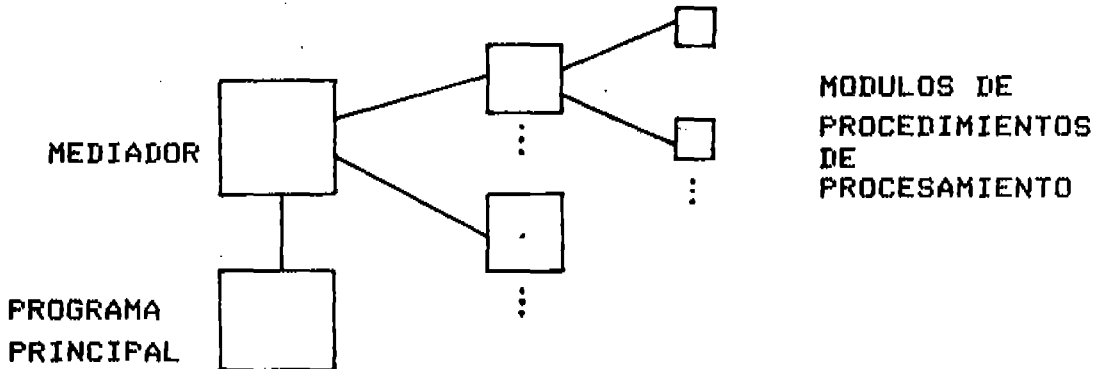


Figura 9.

El diseño usado en SIMASET responde a estas dificultades como sigue:

Está organizado como en el caso anterior, con la diferencia de que el programa principal se divide en dos, apareciendo un mediador donde se incluyen los llamados a todos los módulos del sistema (Figura 9).

SIMASET es el programa principal o huésped con respecto al BINDER del sistema B-6800, al cual en el momento de compilación de algún módulo de una familia de procedimiento se le agrega el código objeto de dicho módulo.

Esto se hace indicando explícitamente que el elemento se compila como SIMASET, es decir, en lugar de enviar el comando a través de CANDE C (por compila el procedimiento independientemente), se debe indicar C AS SIMASET.

El compilar de esta manera a los procedimientos externos de SIMASET permite por un lado economizar archivos en el directorio de archivos del IIMAS, y por consiguiente optimizar el trabajo del administrador de archivos del sistema operativo (pues cada archivo tiene un descriptor que lo identifica de manera única), así como optimizar el proceso de "binding". [2]

También se considera importante mencionar que en el mecanismo de comunicación entre programas se considera a un conjunto de variables globales, las que son usadas por los procedimientos externos al programa principal.

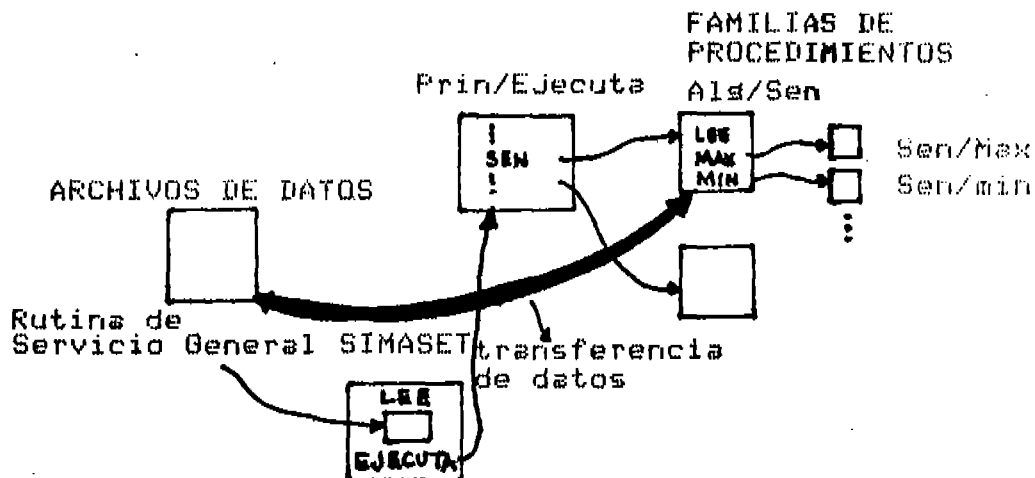
Para el uso de variables globales entre los programas se utiliza el método "INFO FILE METHOD", a través del cual se crea un archivo del tipo "INFO" donde se vacían los contenidos de las tablas principales del programa principal SIMASET, que son generadas por el compilador. [3]

Existen diversos niveles de interrelación entre los subprogramas del sistema con el programa principal: (véase Figura 10)

1) un primer nivel está constituido por la relación entre SIMASET y PRIN/EJECUTA. Como se mencionó anteriormente, el subprograma PRIN/EJECUTA sirve de mediador entre SIMASET y todos los demás subprogramas del sistema.

2) un segundo nivel está constituido por la relación entre PRIN/EJECUTA con todas las raíces de las familias de procedimientos de procesamiento y

3) un tercer nivel de relación está constituido por las raíces de las familias de procedimientos, las que se relacionan con los procedimientos propiamente dichos.



EJEMPLO DE LA INTERCOMUNICACION ENTRE PROGRAMAS

Figura 10.

III.1. PROGRAMAS DE APOYO .

1.1 ARCHIVO1.

El programa ARCHIVO1 tiene la función de generar el archivo que es el directorio de usuarios del sistema DIRUSER. Diruser tiene registros de cuatro palabras, cuyos campos o atributos son:

CLAVE NUMERO SERIE TABLA

Cada uno de estos campos tiene una longitud de una palabra.

Clave: Es la clave que el usuario define la primera vez que ingresa al sistema.

Número: Es el identificador o contraseña, que constituye parte importante para los archivos físicos que genera el usuario.

Serie: El valor inicial bajo este campo es cero, lo que significa que no existen archivos físicos para series; en caso contrario el valor del campo es uno.

Tabla: El valor inicial de este campo es cero, lo que significa que no existen archivos físicos para tablas; en caso contrario el valor del campo es uno.

Diruser es un archivo de tamaño fijo, que consta de 10 registros lo que significa que pueden haber máximo hasta 10 usuarios al sistema en un momento dado.

1.2 VALORES.

Este es un archivo creado en DATA que incluye todas las instrucciones o comandos con que cuenta el sistema.

En él se define el lenguaje que permite establecer la comunicación de un usuario con el sistema; es decir, es un archivo que resume el nivel sintáctico del sistema.

III.2. ADMINISTRADOR DEL SISTEMA.

2.1 SIMASET.

El programa SIMASET, constituye la estructura fundamental del sistema donde se incluyen diversas rutinas de servicio dirigidas a la generación, organización y mantenimiento de los archivos donde se almacena la información.

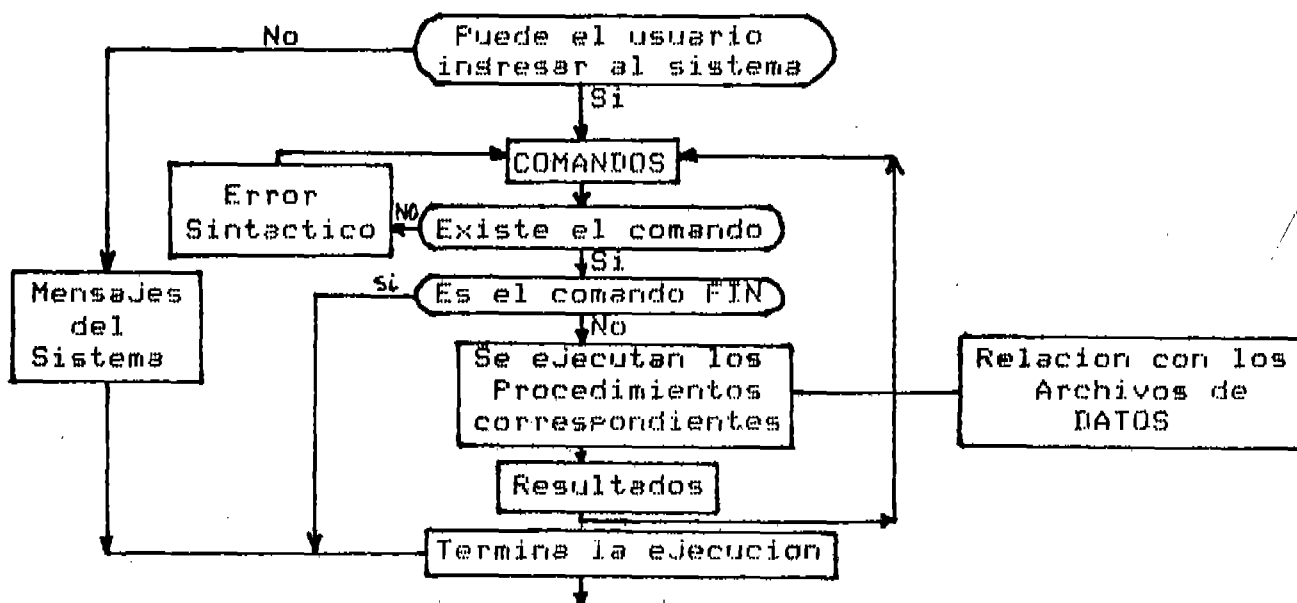
SIMASET a más de tener las rutinas de servicio, incluye la declaración de EJECUTA como un procedimiento externo a él, y lo llama, es decir lo activa. Como se ha mencionado anteriormente EJECUTA a su vez llama a todos los procedimientos del sistema.

La razón fundamental de tener a EJECUTA separado de SIMASET, se debe a que de esta manera existe una flexibilidad en la estructura del sistema que permite agregar tanto nuevas familias de procedimientos, como nuevos procedimientos en las familias definidas, sin tener que realizar procesos muy costosos de computación, sean estos de compilación o del uso del binder.

2.2 PRIN/EJECUTA .

El programa PRIN/EJECUTA a más de realizar los llamados de todos los procedimientos con que cuenta el sistema, permite al usuario generar los cuatro archivos de almacenamiento de la información.

El proceso de intercomunicación del usuario con el sistema, está sintetizado en el siguiente diagrama de flujo, donde se analizan las diversas posibilidades o caminos por donde puede ir dicho proceso.



2.3 COMANDOS DE SIMASET .

Existen dos tipos principales de comandos:

1. Los que se refieren al almacenamiento y recuperación de la información que se denominan COMANDOS DE SERVICIO GENERAL y,
2. Los que se refieren al procesamiento numérico de la información.

Un usuario puede usar el comando ya sea en forma explícita, o una vez que se ha familiarizado con los comandos, los mnemónicos de éstos.

En las secciones 2.3.1 y 2.3.2 se enumeran los comandos con sus correspondientes mnemónicos, y en las secciones 2.3.3 y 2.3.4 se describen las funciones que realizan cada uno de ellos.

2.3.1 COMANDOS DE SERVICIO GENERAL .

Los Comandos de SIMASET que se refieren o actúan sobre la base de datos son:

COMANDO	MNEMONICO
1.- CREASERIE	CES
2.- CREATABLA	CAT
3.- ESCRSERIE	ESE
4.- ESCRTABLA	ETA

5.- LEESERIE	LES
6.- LEETABLA	LAT
7.- LISTSERIE	LIS
8.- LISTTABLA	LIT
9.- ACTUSERIE	ACS
10.- ACTUTABLA	ACT
11.- QITASERIE	QIS
12.- QITATABLA	QIT
13.- TERMINA	TER

2.3.2 COMANDOS DE PROCESAMIENTO NUMERICO DE LA INFORMACION .

Los comandos de SIMASET que se refieren al procesamiento numérico de la información son :

COMANDO	MNEMONICO
-----	-----
1.- SEN/1(MAX)	SN1
2.- SEN/2(MIN)	SN2
3.- SEN/3(MEDIA)	SN3
4.- SEN/4(NORM2)	SN4
5.- SEN/5(NORMINF)	SN5
6.- SEN/6(NORM1)	SN6
7.- SEN/7(DESUIA)	SN7
8.- SES/1(SUMA)	SE1
9.- SES/2(MULT)	SE2
10.- SES/3(COMVOL)	SE3
11.- TAT/1(SUMA)	TT1
12.- TAT/2(PREDICT)	TT2
13.- TAT/3(CONTROL)	TT3
14.- TAN/1(MAX)	TN1
15.- TAN/2(MIN)	TN2
16.- TAN/3(MEDIA)	TN3
17.- SS/1(FILTRA)	S11
18.- SS/2(FILTRA)	S12
19.- SS/3(FILTRA)	S13
20.- SS/1(ESPECTRO)	S21
21.- SS/2(ESPECTRO)	S22
22.- SS/3(ESPECTRO)	S23
23.- SS/1(SUAVIZA)	S31
24.- SS/2(SUAVIZA)	S32
25.- SS/3(SUAVIZA)	S33
26.- SS/(ESCALAR)	SSE
27.- SET/1(IDENT)	ST1
28.- SET/2(IDENT)	ST2
29.- SET/3(IDENT)	ST3
30.- SET/1(HISTO)	ST4
31.- SET/2(HISTO)	ST5
32.- SET/3(HISTO)	ST6
33.- TAS/1(SIMULA)	TS1

34.- TAS/2(SIMULA)	TS2
35.- TAS/3(SIMULA)	TS3
36.- GRA/1(CONT)	GR1
37.- GRA/2(HISTO)	GR2
38.- GRA/3(HISTO)	GR3
39.- GRA/4(LINEAS)	GR4
40.- GRA/5(PUNTOS)	GR5

Todos estos comandos están implementados como rutinas independientes, y se refieren a los programas del tercer tipo, es decir a los programas que sirven para el procesamiento de la información.

2.3.3 DESCRIPCION DE LOS COMANDOS DE SERVICIO GENERAL .

Es muy importante el hacer una descripción detallada de cada uno de los Comandos que actúan sobre la base de datos.

2.3.3.1 CREASERIE .

El mnemónico del comando es CES.

Este comando crea tanto el archivo físico en disco que se utiliza para almacenar las diferentes series que tiene un usuario, como el archivo físico que sirve de directorio para las series.

Las acciones de este comando son:

a) Busca en el archivo DIRUSER el registro que tiene la clave del usuario, y verifica si el valor que está en el atributo serie del registro es o no cero.

b) Si el valor de SERIE en DIRUSER es cero, genera estos archivos físicos, cambia el valor de SERIE a uno. Posteriormente el sistema envía la siguiente información:

* LOS ARCHIVOS FISICOS PARA SERIES SON :
 {clave}/ARCH1/{id}
 {clave}/DIRECTORIO/{id}

w termina la operación con dos datos (**)

c) Si el valor de SERIE en DIRUSER ya es uno, el sistema envía el mensaje al usuario:

{* YA CREO SU ARCHIVO DE SERIES *}

terminando con dos satos.

##

2.3.3.2 CREATABLA .

El mnemónico del comando es CAT.

Este comando es muy similar a CREASERIE, con la diferencia de que los archivos físicos que se generan en disco se refieren por un lado a las tablas que va a almacenar el usuario y por otro al directorio de estas tablas.

Las acciones de este comando son:

- a) Busca en el archivo DIRUSER el registro que tiene la clave del usuario, y verifica si el valor que está en el atributo TABLA del registro es o no cero.
- b) Si el valor de TABLA en DIRUSER es cero, genera estos archivos físicos, cambia el valor de TABLA a uno, el sistema envía al usuario la siguiente información:

* LOS ARCHIVOS FISICOS PARA TABLAS SON :
 {clave}/TABLA/{id}
 {clave}/DIRTAB/{id}

y termine la operación con dos satos (##).

- c) Si el valor de TABLA en DIRUSER ya es uno, el sistema envía el mensaje al usuario:

(* YA CREO SU ARCHIVO DE TABLAS *)

terminando con dos satos.

##

2.3.3.3 ESCRSERIE .

El mnemónico del comando es ESE.

Este comando permite almacenar en disco los datos que tiene el usuario y que se refieren a una serie.

Realize las siguientes acciones:

- a) Si el usuario no ha creado su archivo físico para series, el comando envía el mensaje:

(* DEBIO CREAR ARCHIVO PREVIAMENTE *)

terminando la ejecución del comando.

##

- b) Si existe el archivo físico para series, el comando envía el

mensaje:

{* DE UN MAXIMO DE 6 CARACTERES PARA
EL NOMBRE DEL ARCHIVO A CREAR *}

El usuario entonces debe proceder a nombrar la serie de datos que quiere almacenar en disco (por ej. DATOS1).

c) Ahora bien, se busca en el archivo de directorio de series (DIRECTORIO), si ese nombre ya fue usado previamente en otra serie. En caso de haber sido usado, el sistema envía el mensaje:

{* YA EXISTE ESE ARCHIVO *}
{* DE UN NUEVO NOMBRE DE ARCHIVO *}

repetiéndose la operación anterior.

d) De otra manera, el sistema envía el mensaje:

{* ESCRIBA SUS DATOS *}

entonces el usuario debe escribir sus datos muy cuidadosamente (uno por línea).

En caso de que el usuario se percatara de que cometió un error en el dato actual, puede usar la tecla DELETE del teletipo, y teclear nuevamente el dato.

Al término de esta operación el usuario debe escribir los cuatro caracteres:

?END

Una vez que el usuario ha terminado de escribir todos los datos de la serie que quiere almacenar en disco, se almacenan junto a los datos, el número de serie, el nombre y el tamaño en el archivo de series ARCH1, y a más de esto se almacena la dirección final del último dato de este sub-archivo en el archivo DIRECTORIO.

e) Luego el sistema envía el mensaje:

{* DESEA CONOCER EL TAMANO DE LA SERIE *}

si el usuario contesta SI, se da este tamaño, en caso contrario el sistema envía el mensaje:

{* EN CASO DE QUE ESCRIBIO EN DISCO
DESEA UN LISTADO?CONTESTE SI O NO *}

si el usuario contesta SI, el sistema envía el mensaje:

{* ESPECIFIQUE EL INTERVALO *}

ejecutándose la operación LISTSERIE que se describirá más adelante.

Si el usuario contesta NO, termina el comando.

##

2.3.3.4 ESCRTABLA .

El mnemónico del comando es ETA.

Este comando permite que el usuario almacene en disco los datos de tipo tabla.

Realiza las siguientes acciones:

a) Si el usuario no ha creado su archivo físico para tablas, el comando envía el mensaje:

{* DEBIO CREAR ARCHIVO PREVIAMENTE *}

terminando la ejecución del comando.

b) Si existe el archivo físico para tablas, el comando envía el mensaje:

{* DE UN MAXIMO DE 6 CARACTERES PARA
EL NOMBRE DEL ARCHIVO A CREAR *}

El usuario entonces debe proceder a nombrar el conjunto de datos de tipo tabla que quiere almacenar en disco (por ej. TABLA1).

c) Se busca en el archivo de directorio de tablas (DIRTAB), si ese nombre fue usado previamente en otra tabla. En caso de haber sido usado, el sistema envía el mensaje:

{* YA EXISTE ESE ARCHIVO *}
{* DE UN NUEVO NOMBRE DE ARCHIVO *}

repetiéndose la operación anterior.

d) Si el nombre con el que va a almacenar los datos no existe, el sistema envía los mensajes:

{* CUAL ES EL ORDEN DE LA TABLA *}

el usuario debe definir el orden N.

{* DEFINA EL NUMERO DE COLUMNAS *}

refiriéndose al número P de la definición de tabla.

{* DEFINA EL NUMERO DE FILAS *}

es el número M. Cuando se obtenga el mensaje :

{* ESCRIBA SUS DATOS *}

entonces el usuario debe escribir un total de $(N+1)*M*P$ datos (uno por línea). Los primeros P datos se refieren a la primera fila y así sucesivamente.

Una vez que el usuario ha terminado de escribir todos los datos de la tabla que quiere almacenar en disco, se almacenan junto a los datos el número de la tabla, el nombre y el tamaño en el archivo de tablas TABLA, y a más de esto la dirección final del último dato del sub-archivo en el archivo DIRTAB.

e) Luego el sistema envía el mensaje:

```
(* EN CASO DE QUE ESCRIBIO EN DISCO
DESEA UN LISTADO? CONTESTE SI O NO *)
```

Si el usuario contesta SI, el sistema envía el mensaje:

```
(* ESPECIFIQUE EL INTERVALO EN N *)
```

teniendo que explicitar el límite inferior y el límite superior del conjunto de matrices que desea listar, ejecutándose la operación LISTTABLA que se describirá más adelante, terminando la ejecución del comando.

##

2.3.3.5 LEESERIE .

El mnemónico del comando es LES.

Este comando permite leer los datos de tipo SERIE que están almacenados en disco, y cargarlos en un arreglo de memoria principal.

Realiza las siguientes acciones:

a) Verifica si el usuario creó su archivo físico para series. En caso de no haberlo hecho, el sistema envía el mensaje:

```
(* DEBIO CREAR ARCHIVO PREVIAMENTE *)
```

terminando la ejecución del comando.

##

b) El sistema envía el mensaje :

```
(* QUE ARCHIVO QUIERE LEER *)
```

el usuario debe dar explícitamente el nombre del archivo (que en realidad es un sub-archivo del archivo ARCH1) del cual va a leer los datos. Ese nombre se busca en el directorio de series (DIRECTORIO). Si existe, procede a leer todos los datos de la serie que se encuentran en el sub-archivo correspondiente. Si no existe, el sistema envía el mensaje:

```
(* NO EXISTE ESE ARCHIVO EN DISCO *)
```

y termina la ejecución del comando.

##

c) Luego el sistema envía el mensaje:

{* DESEA CONOCER EL TAMANO DE LA SERIE *}

Si el usuario contesta SI, se le da a conocer ese tamaño, de otra manera el sistema envía el mensaje:

{* DESEA UN LISTADO? CONTESTE SI O NO *}

En caso de que el usuario conteste SI, el sistema envía el mensaje:

{* ESPECIFIQUE EL INTERVALO *}

teniendo el usuario que indicar los límites inferior y superior de los datos de la serie, y se ejecuta el comando LISTSERIE. Si el usuario contesta NO, termina el comando.

##

2.3.3.6 LEETABLA .

El mnemónico del comando es LAT.

Este comando permite leer los datos de tipo tabla que están almacenados en disco, y cargarlos en un arreglo de memoria principal.

Realiza las siguientes acciones:

a) Verifica si el usuario creó el archivo físico para tablas. En caso de que no lo hizo el sistema envía el mensaje:

{* DEBIO CREAR ARCHIVO PREVIAMENTE *}

terminando la ejecución del comando.

##

b) El sistema envía el mensaje:

{* QUE ARCHIVO QUIERE LEER *}

el usuario debe dar el nombre del archivo del que quiere leer sus datos. Ese nombre se busca en el directorio de tablas (DIRTAB). Si existe el nombre, procede a leer todos los datos del archivo TABLA. Si no existe, el sistema envía el mensaje:

{* NO EXISTE ESE ARCHIVO EN DISCO *}

y termina la ejecución del comando.

##

c) El sistema envia el mensaje:

{* DESEA CONOCER EL ORDEN DE SU TABLA *}

si el usuario contesta SI se le da a conocer el orden N de la tabla, de otra manera, el sistema envia el mensaje:

{*DESEA UN LISTADO? CONTESTE SI O NO *}

En caso de que el usuario contesta SI, el sistema envia el mensaje:

{* ESPECIFIQUE EL INTERVALO EN N *}

teniendo el usuario que indicar los limites inferior y superior de las matrices que quiere listar, y se ejecuta la operacion LISTTABLA, terminando la ejecucion del comando. Si el usuario contesta NO, termina la ejecucion del comando.

##

2.3.3.7 LISTSERIE .

El mnemónico del comando es LIS.

Este comando se ejecuta solamente después de que se ha ejecutado alguno de los comandos, como:

ESCRSERIE
LEESERIE
ACTUSERIE

siendo parte de estos, es decir la rutina que realiza el listado de los elementos de una tabla está llamada por los comandos previos. Si el usuario al ejecutar alguno de los comandos anteriores especificó los limites del intervalo de los datos de la serie que quiere listar, se cumplen las siguientes acciones:

a) Si hubo algún error en los limites del intervalo, por ejemplo: se excedió del tamaño de la serie, o comenzó desde 0, es decir el límite inferior es 0 (el límite inferior no debe ser menor que 1), el sistema envia el mensaje :

{* ERROR EN LOS LIMITES DEL INTERVALO *}

le dá el tamaño de la serie, y termina la ejecución del comando.

##

b) En caso contrario, el sistema envia el mensaje:

{* A CONTINUACION SE LISTAN LOS DATOS
ORIGINALES DE DISCO *}

procediéndose a listar tanto la posición en la que se encuentra el

dato, como el dato de la serie. Una vez se llega al límite superior del intervalo, termina la ejecución del comando.

##

2.3.3.8 LISTTABLA .

El mnemónico del comando es LIT.

Este comando se ejecuta dentro de los comandos:

ESCTABLA
LEETABLA
ACTUTABLA

Sus acciones son:

a) Una vez que se han especificado los límites inferior y superior del intervalo en N (es decir del orden), el sistema envía el mensaje:

{* A CONTINUACION SE LISTAN LOS DATOS
ORIGINALES DE DISCO *}

luego sale el letrero:

LOS COEFICIENTES DE N= SON

y se listan los datos en forma matricial, en la que aparece un indicador como R1., R2., ETC, significando que se trata del renglón 1, renglón 2, etc, y previo al dato la posición que ocupa en el renglón, es decir la columna. Como se ha establecido antes, las matrices son de dimensiones MXP. Así mismo se ha establecido que existen N+1 matrices lo que significa que el orden en N varía de 0 a N, y termina la ejecución del comando.

##

2.3.3.9 ACTUSERIE .

El mnemónico del comando es ACS.

Este comando permite modificar los datos de la serie que se encuentran almacenados en disco.

Realiza las siguientes acciones:

a) El sistema envía el mensaje:

{* CUAL ARCHIVO VA A ACTUALIZAR *}

El usuario debe especificar el nombre del archivo a actualizar.

En caso de que no exista esa serie almacenada en disco, el sistema envía el mensaje:

{* NO EXISTE ESE ARCHIVO EN DISCO *}

terminando la ejecución del comando.

##

b) Se busca en el archivo directorio de series DIRECTORIO, si existe o no el nombre de la serie, dado por el usuario, si existe, el sistema envía el mensaje:

{* ESCRIBA: POSICION,DATO,NUEVODATO *}

el usuario entonces debe escribir por el periférico, en la misma línea y separado por comas: la posición en la que se encuentra el dato que quiere cambiar, el valor del dato que está almacenado en disco, y el nuevo valor que va a ser cambiado y almacenado en disco. Si el usuario comete un error, el sistema envía el mensaje y la instrucción:

{* ERROR EN EL DATO ALMACENADO *}

EL DATO EN "POSICION" ES "VALOR DEL DATO"

Procediendo a realizar la misma pregunta anterior.

c) Luego de que ha realizado la modificación del dato almacenado, el sistema envía el mensaje:

{* DESEA UN LISTADO? CONTESTE SI O NO *}

si el usuario contesta SI, el sistema envía el mensaje:

{* ESPECIFIQUE EL INTERVALO *}

teniendo el usuario que especificar los límites inferior y superior de los datos almacenados en disco que quiere listar, ejecutándose la operación LISTSERIE, y terminando la ejecución del comando.

##

2.3.3.10 ACTUTABLA .

El mnemónico del comando es ACT.

Este comando realiza las mismas acciones que el comando ACTUSERIE en a y b, con la diferencia de que las modificaciones se realizan en el archivo físico de tablas TABLA, y en lugar de escribir: posición, dato, nuevodato, el usuario debe escribir en la misma línea y separado por comas:

MATRIZ DONDE ESTA EL DATO,RENGLON DE LA MATRIZ,COLUMNA DE LA MATRIZ,EL DATO ACTUAL ALMACENADO,Y EL NUEVO DATO.

si el usuario comete algún error, el sistema envía el mensaje, y la instrucción:

{* ERROR EN EL DATO ALMACENADO *}

EN LA MATRIZ ---

RENGLON ---

COLUMNA ---
EL VALOR DEL DATO ES ---

y nuevamente realiza la operación anterior.

c) Luego de que ha realizado la modificación del dato almacenado, el sistema envía el mensaje:

{* DESEA UN LISTADO? CONTESTE SI O NO *}
si el usuario contesta SI, el sistema envía el nuevo mensaje:

{* ESPECIFIQUE EL INTERVALO EN N *}

teniendo el usuario que dar los límites inferior y superior del orden N. Si el usuario contesta NO a alguno de los mensajes previos, termina el comando.

##

2.3.3.11 QITASERIE .

El mnemónico del comando es QIS.

Este comando suprime o elimina tanto del archivo de series ARCHI, como del directorio de series DIRECTORIO, toda la información que está relacionada con la serie que el usuario ya no necesita.

Realiza las siguientes acciones:

a) Verifica si el usuario creó los archivos físicos para series. En caso de que no lo hizo el sistema envía el mensaje:

{* DEBIO CREAR ARCHIVO PREVIAMENTE *}

terminando la ejecución del comando.

##

b) El sistema envía el mensaje:

{* CUAL ARCHIVO DESEA SUPRIMIR *}

El usuario debe dar el nombre de la serie que quiere eliminar del archivo de series.

Si no existe esa serie, el sistema envía el mensaje:

{* NO EXISTE ESE ARCHIVO EN DISCO *}

y termina la ejecución del comando.

##

c) En caso contrario, es decir si existe la serie, se la busca en el directorio de series, y el nombre particular de la serie se reemplaza por "DISPON".

d) Se leen todos los registros del directorio de series DIRECTORIO,

hasta encontrar el registro en el que el valor del campo o atributo NOMBRE sea "DISPON". Una vez encontrado este registro se procede a cargar todas las series que están a continuación de ésta, una por una en memoria principal, y de ahí se escriben de regreso en el archivo físico. De otra manera no se realiza ninguna acción.

Cabe indicar que se reactualizan las direcciones y los apuntadores en el archivo DIRECTORIO de todas las series (que como se mencionó anteriormente son sub-archivos del archivo ARCH1), apareciendo en el directorio de series el registro "DISPON 0 0", como su último registro.

Por último el sistema envía el mensaje:

* SE ELIMINA EL ARCHIVO *
NOMBRE

donde NOMBRE es el nombre de la serie que se elimina, y termina el comando.

##

2.3.3.12 QITATABLA .

El mnemónico del comando es QIT.

Este comando realiza exactamente las mismas operaciones, que el comando QITASERIE, con la diferencia que actúa sobre los archivos de tablas TABLA y DIRTAB.

2.3.3.13 TERMINA .

El mnemónico del comando es TER.

Este comando elimina todos los archivos físicos de un usuario, así como se lo elimina del archivo de usuarios DIRUSER, concluyendo su participación en el sistema SIMASET.

Realiza las siguientes acciones:

a) Busca la clave del usuario, en el directorio de usuarios DIRUSER. Una vez la encuentra determina si la contraseña es correcta, entonces cambia el nombre de la clave por la sílaba "CAMB", cambiando también los otros valores de los atributos del registro por ceros.

b) El sistema envía el mensaje:

(* SE ELIMINA DE LA LISTA DE USUARIOS *)

Si el usuario tenía archivos de series, se eliminan estos archivos, y el sistema envía el mensaje:

(* SE ELIMINAN SUS ARCHIVOS DE SERIES *)

Si el usuario tenía archivos de tablas, se eliminan estos archi-

vos físicos, y el sistema envía el mensaje:

{* SE ELIMINAN SUS ARCHIVOS DE TABLAS *}

y termina el comando.

##

2.3.4 DESCRIPCION DE LOS COMANDOS DE PROCESAMIENTO NUMERICO DE LA INFORMACION .

Las instrucciones o comandos que se refieren al procesamiento numérico de la información son :

2.3.4.1 SEN/1(MAX) .

El mnemónico del comando es SN1.

Este comando obtiene el máximo absoluto de un conjunto de datos del tipo SERIE.

El padre de este algoritmo es ALG/SEN, y es ahí donde se ejecutan verdaderamente las acciones que realiza el comando.

Realiza las siguientes acciones :

a) El sistema envía los mensajes :

- * EL COMANDO OBTIENE EL MAXIMO VALOR GLOBAL DE SUS DATOS *
- * EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la información :

```

-----
EL VALOR MAXIMO ES
      DATO [?] ---
-----

```

donde (?), es el lugar con respecto a la secuencia de la serie en la que está dicho elemento, terminando la ejecución del comando.

##

2.3.4.2 SEN/2(MIN) .

El mnemónico del comando es SN2.

Este comando obtiene el elemento que es el mínimo absoluto de un conjunto de datos del tipo serie.

Realiza las siguientes acciones :

a) El sistema envía los mensajes :

* EL COMANDO OBTIENE EL MINIMO
VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la información :

EL VALOR MINIMO ES
DATO [?] ---

donde (?), es el lugar en el que se localiza el dato, y termina la ejecución del comando.

**

2.3.4.3 SEN/3(MEDIA) .

El mnemónico del comando es SN3.

Este comando obtiene la media del conjunto de datos del tipo SERIE.

El padre de este algoritmo es ALG/SEN. Realiza las siguientes acciones :

a) El sistema envía los mensajes :

* EL COMANDO OBTIENE LA MEDIA DE LOS
DATOS ALMACENADOS EN DISCO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la información:

LA MEDIA DE SUS DATOS ES

y termina la ejecución del comando.

**

2.3.4.4 SEN/4(NORM2) .

El mnemónico del comando es SN4.

Este comando obtiene la norma-2 del conjunto de datos del tipo SERIE, los cuales pueden ser tratados como un vector.

Realiza las siguientes acciones :

a) El sistema envía los mensajes :

* EL COMANDO OBTIENE LA NORMA 2 *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envia la información :

LA NORMA 2 DE LA SERIE
ES ---

terminando la ejecución del comando.

**

2.3.4.5 SEN/5(NORMINF) .

El mnemónico del comando es SN5.

Este comando obtiene la norma infinito del conjunto de datos del tipo serie.

Realiza las siguientes acciones :

a) El sistema envia los mensajes :

* EL COMANDO OBTIENE LA NORMA INFINITO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

Procediéndose a la ejecución de LEESERIE.

b) El sistema envia la información :

LA NORMA INFINITO DE LA SERIE
ES ---

y termina la ejecución del comando.

**

2.3.4.6 SEN/6(NORM1) .

El mnemónico del comando es SN6.

Este comando obtiene la norma-1 del conjunto de datos del tipo SERIE.

Realiza las siguientes acciones :

a) El sistema envia los mensajes :

* EL COMANDO OBTIENE LA NORMA 1 *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

Procediéndose a la ejecución de LEESERIE.

b) El sistema envia la información :

LA NORMA 1 DE LA SERIE
ES ---

terminando la ejecución del comando.

**

2.3.4.7 SEN/7(DESUIA) .

El mnemónico del comando es SN7.

Este comando obtiene la desviación Standard del conjunto de datos del tipo SERIE.

Realiza las siguientes acciones:

a) El sistema envía los mensajes :

* EL COMANDO OBTIENE LA DESVIACION STANDARD *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) el sistema envía la información :

LA DESVIACION STANDARD DE SUS DATOS ES

terminando la ejecución del comando.

**

2.3.4.8 SES/1(SUMA) .

El mnemónico del comando es SE1.

Este comando suma los valores de los elementos de dos series que están almacenadas en disco.

Sean S1 y S2 dichas series, entonces :

$$s1(i)+s2(i) = (s1+s2)(i)$$

donde el índice i varía de 1 al máximo valor que existe entre los tamaños de S1 y S2.

Realiza las siguientes acciones:

a) El sistema envía los mensajes :

* EL COMANDO SUMA LOS VALORES DE
LAS SERIES ALMACENADAS EN DISCO *
* EJECUTA DOS VECES EL COMANDO
LEESERIE *

procediéndose a la ejecución del comando LEESERIE. Esta se realiza por dos veces consecutivas, debido a que es necesario cargar en memoria principal, el contenido de todos los elementos de las dos series.

b) Una vez están listas las series, se realiza la suma de los elementos correspondientes.

El sistema envía el mensaje :

* DESEA CONOCER SUS RESULTADOS *

si el usuario contesta SI, el sistema envía el nuevo mensaje por el

periférico :

*** ESPECIFIQUE EL INTERVALO ***

teniendo el usuario que escribir los límites inferior y superior de los datos. Si comete algún error en la operación anterior, el sistema envía el mensaje :

***** ERROR EN EL INTERVALO *****
EL TAMAÑO DE LA SERIE VARIA DE 1 A ---

repetiéndose la operación.

c) Por último, existe la posibilidad de almacenar los nuevos datos obtenidos en disco, ejecutándose para esto el comando ESCRSERIE. El sistema envía el mensaje:

*** DESEA ALMACENAR SUS NUEVOS VALORES ***

si el usuario contesta SI, se ejecuta ESCRSERIE, que ya fue descrito anteriormente. En caso contrario, termina la ejecución del comando.

##

2.3.4.9 SES/2(MULT) .

El mnemónico del comando es SE2.

Este comando multiplica los valores de los elementos de dos series que están almacenadas en disco.

Sean S1 y S2 dichas series y sean $s1(i)$ elemento de S1, $s2(i)$ elemento de S2, entonces :

$s1(i) * s2(i) = (s1*s2)(i)$
 $i=1,2,\dots,\text{Max}(\text{tamaño de S1}, \text{tamaño de S2}).$

Realiza las siguientes acciones:

a) El sistema envía los mensajes :

*** EL COMANDO MULTIPLICA LOS VALORES DE
LAS SERIES ALMACENADAS EN DISCO ***
*** EJECUTA DOS VECES EL COMANDO
LEESERIE ***

entonces, el usuario se relaciona por dos veces consecutivas con el comando LEESERIE.

b) Una vez están listas las series, se realiza la multiplicación de los elementos correspondientes.

El sistema envía el mensaje :

*** DESEA CONOCER SUS RESULTADOS ***

si el usuario contesta SI, el sistema envía el nuevo mensaje:

*** ESPECIFIQUE EL INTERVALO ***

teniendo el usuario que escribir los límites inferior, y superior de los datos. Si comete algún error en la operación, el sistema envía el mensaje:

***** ERROR EN EL INTERVALO *****

EL TAMANO DE LA SERIE VARIA DE 1 A ---

repetiéndose la operación.

c) El sistema envía el mensaje :

*** DESEA ALMACENAR SUS NUEVOS VALORES ***

si el usuario contesta SI, se ejecuta el comando ESCRSERIE. En caso contrario termina el comando.

##

2.3.4.10 SES/3(CONVOL) .

El mnemónico del comando es SE3.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo simula las diferentes acciones previas a la convolución propiamente dicha, esto es :

a) El sistema envía los mensajes :

*** EL COMANDO REALIZA LA CONVOLUCION ***

*** EJECUTA DOS VECES EL COMANDO**

LEESERIE *

teniendo el usuario que relacionarse por dos veces consecutivas con el comando LEESERIE.

b) Las series están listas para realizar la operación de convolución, y el sistema envía la información, :

*** OBSERVACION :**

ALGORITMO DE CONVOLUCION NO IMPLEMENTADO.
EL PROGRAMADOR DEBE TOMAR EN CUENTA QUE LA
RUTINA OPERA CON DOS SERIES, LA UNA ESTA EN
EL ARREGLO A[I1:TAM1], LA OTRA EN EL ARRE-
GLO B[I1:TAM2]. ASI MISMO LOS RESULTADOS
OBTENIDOS DEBEN SER GUARDADOS EN EL ARRE-
GLO C[III], DONDE I DEBE VARIAR DE 1 AL
MAX(TAM1,TAM2). *

c) Luego el sistema envía el mensaje:

*** DESEA ALMACENAR SUS NUEVOS VALORES ***

en este caso, hasta que no esté implementado el algoritmo, el usuario debe contestar NO, terminando la ejecución del comando. Una vez implementada la rutina, y si el usuario contesta SI a la pregunta anterior, se ejecuta el comando ESCRSERIE.

##

2.3.4.11 TAT/1(SUMA) .

El mnemónico del comando es TT1.

Este comando suma los valores de los elementos de dos tablas que están almacenadas en disco.

Sean las tablas T1 y T2, tales que: $t1(i)$ sea un elemento de T1 y $t2(i)$ sea un elemento de T2, entonces

$$t1(i) + t2(i) = (t1+t2)(i)$$

En este caso la suma se realiza sobre matrices, por lo que esta operación se puede realizar solo y solo si las dimensiones de las matrices consideradas son iguales.

Realiza las siguientes acciones:

a) El sistema envía los mensajes:

* EL COMANDO SUMA LOS VALORES DE
LAS TABLAS ALMACENADAS EN DISCO *
* EJECUTA EL COMANDO LEETABLA *

el comando LEETABLA se ejecuta por dos veces consecutivas, para cargar las tablas en memoria principal.

b) Una vez están listas las tablas, y siempre y cuando sus elementos, es decir las matrices tengan las mismas dimensiones, se realiza la operación de suma. De otra manera el sistema envía el mensaje:

* ERROR EN LAS DIMENSIONES DE LAS TABLAS *

terminando el comando.

##

c) Si se realizó la suma de las tablas, el sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

si el usuario contesta SI, el sistema envía la información :

* OBSERVACION :
EL ORDEN DE SU NUEVA TABLA ES ---
EL NUMERO DE COLUMNAS ES ---
EL NUMERO DE FILAS ES ---

teniendo el usuario que escribir estos valores una vez se ejecute el comando ESCR TABLA. Si el usuario comete algún error en esta operación, lo más probable es que obtenga un error de INDICE INVALIDO, terminando la ejecución no solo del comando, sino del sistema en general. Por lo que se recomienda que el usuario sea muy cui-

dadoso en esta operación. En caso contrario, es decir si el usuario contesta NO al mensaje anterior, termina el comando sin realizar ninguna operación de escritura.

##

2.3.4.12 TAT/2(PREDICT) .

El mnemónico del comando es TT2.

Este comando determina los coeficientes F_s y U_s del predictor.

[4]

Para determinar estos, se relacionan a los elementos de dos conjuntos de tablas que están almacenadas en disco.

Sean A y B dichas tablas, entonces a partir de éstas y basándose en las ecuaciones:

$$b_1 = f_1 + a_1$$

$$b_2 = f_2 + a_1 \cdot f_1 + a_2$$

$$b_3 = f_3 + a_1 \cdot f_2 + a_2 \cdot f_1 + a_3$$

.

.

.

$$b(k-1) = f(k-1) + a_1 \cdot f(k-2) + a_2 \cdot f(k-3) + \dots + a(k-2) \cdot f_1 + a(k-1)$$

$$b_k = u_0 + a_1 \cdot f(k-1) + a_2 \cdot f(k-2) + \dots + a(k-1) \cdot f_1 + a_k$$

$$b(k+1) = u_1 + a_2 \cdot f(k-1) + a_3 \cdot f(k-2) + \dots + a_k \cdot f_1 + a(k+1)$$

.

.

.

$$b_n = u(n-k) + a(n-k+1) \cdot f(k-1) + \dots + a(n-1) \cdot f_1 + a_n$$

$$0 = u(n-k+1) + a(n-k+2) \cdot f(k-1) + \dots + a(n-1) \cdot f_1 + a_n$$

$$0 = u(n-k+2) + a(n-k+3) \cdot f(k-1) + \dots + a_n \cdot f_1$$

$$0 = u(n-k+3) + a(n-k+4) \cdot f(k-1) + \dots + a_n \cdot f_2$$

.

.

.

$$0 = u(n-2) + a(n-1) \cdot f(k-1) + a_n \cdot f(k-2)$$

$$0 = u(n-1) + a_n \cdot f(k-1)$$

se determinan los valores para las F_s y las U_s .

Realiza las siguientes acciones:

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE LOS COEFICIENTES F_S Y U_S DEL
PREDICTOR *

* EJECUTA EL COMANDO LEETABLA *

el comando LEETABLA se ejecuta por dos veces consecutivas con el propósito de cargar en memoria principal las tablas que van a estar en los arreglos A y B.

b) Una vez están listas las tablas, y siempre y cuando el número de

columnas y el número de filas de las matrices sean uno (es decir en realidad las tablas almacenadas, en este caso son vectores), se realiza la operación.

De otra manera el sistema envía el mensaje :

* NO SE PUEDEN CALCULAR LAS FS Y US DEL PREDICTOR *

terminando la ejecución del comando.

##

c) El sistema envía el mensaje :

* DESEA CONOCER SUS RESULTADOS *

si el usuario contesta SI, el sistema envía la información :

```
-----  
LOS VALORES DE F SON  
  F[1] ---  
  F[2] ---  
  .  
  .  
  .  
LOS VALORES DE U SON  
  U[0] ---  
  U[1] ---  
  .  
  .  
  .  
-----
```

d) Por último el sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

si el usuario contesta SI, el sistema envía la información :

* OBSERVACION :

```
EL ORDEN DE SU NUEVA TABLA ES ---  
EL NUMERO DE COLUMNAS ES ---  
EL NUMERO DE FILAS ES ---
```

Se ejecuta el comando ESCR TABLA, y termina el comando.

##

2.3.4.13 TAT/3(CONTROL) .

El mnemónico del comando es TT3.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

*** EL COMANDO REALIZA EL CONTROLADOR *
* EJECUTA EL COMANDO LEETABLA ***

el comando LEETABLA se ejecuta por dos veces consecutivas, con el propósito de cargar en memoria principal a las tablas que se quieren relacionar.

b) Una vez están listas las tablas, el sistema envía las instrucciones: OBSERVACION1 que está localizada físicamente en el archivo ALG/TAT (10100-10600), y OBSERVACION2 que está localizada físicamente en el archivo TAT/CONTROLA.

*** OBSERVACION 1:**

EL PROGRAMADOR DEBE DEFINIR LAS CONDICIONES INICIALES NECESARIAS EN EL ALGORITMO PADRE ALG/TAT, PARA ESTO PUEDE HACER UN LISTADO DE LA 7400-9200 DE ALG/TAT, QUE SE REFIERE A COMO SE IMPLEMENTO PARA EL CASO DEL PREDICTOR. *

*** OBSERVACION 2:**

ALGORITMO CONTROLA NO IMPLEMENTADO.
EL PROGRAMADOR DEBE SABER QUE ESTA ACTUANDO CON DOS CONJUNTOS DE TABLAS: LA UNA ESTA EN EL ARREGLO A1[1:TAM1] (A1[0]=1); Y LA OTRA ESTA EN B1[1:TAM2] (B1[0]=1). SI SE DESEAN ALMACENAR LOS RESULTADOS DEBE USARSE EN LA RUTINA EL ARREGLO CCCC11, DONDE EL TAMAÑO DEL ARREGLO DEBE SER ACTUALIZADO CON UN RESIZE, Y EL INDICE I VARIA DE 1 AL MAX(TAM1,TAM2). *

c) El sistema envía el mensaje:

*** DESEA ALMACENAR SUS NUEVOS VALORES ***

el usuario debe contestar NO, hasta que el algoritmo haya sido implementado, y termina el comando.

NOTA: Una vez implementado el algoritmo debe el analista modificar las líneas de 10100-10600 del padre ALG/TAT, y substituir por las condiciones iniciales que deben cumplir las tablas para realizar el CONTROLADOR.

##

2.3.4.14 TAN/1(MAX) .

El mnemónico del comando es TN1.

Este comando obtiene el máximo absoluto de un conjunto de datos del tipo TABLA.

El padre de este algoritmo es ALG/TAN. Realiza las siguientes acciones:

a) El sistema envía los mensajes:

*** EL COMANDO OBTIENE EL MAXIMO**

VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEETABLA *

procediéndose a la ejecución de LEETABLA.

b) El sistema envía la información :

EL VALOR MAXIMO ESTA EN LA MATRIZ:
ORDEN N ----
REGLON M ----
COLUMNA P ----
Y ES ----

y termina la ejecución del comando.

##

2.3.4.15 TAN/2(MIN) .

El mnemónico del comando es TN2.

Este comando obtiene el mínimo absoluto de un conjunto de datos del tipo TABLA.

El padre del algoritmo es ALG/TAN.

Realiza las siguientes acciones:

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE EL MINIMO
VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEETABLA *

procediéndose a la ejecución de LEETABLA.

b) El sistema envía la información:

EL VALOR MINIMO ESTA EN LA MATRIZ:
ORDEN N ----
REGLON M ----
COLUMNA P ----

y termina la ejecución del comando.

##

2.3.4.16 TAN/3(MEDIA) .

El mnemónico del comando es TN3.

Este comando obtiene la media de un conjunto de datos del tipo TABLA.

El padre del algoritmo es ALG/TAN. Realiza las siguientes acciones:

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE LA MEDIA DE LOS
 DATOS ALMACENADOS EN DISCO *
 * EJECUTA PRIMERO EL COMANDO LEETABLA *

procediéndose a la ejecución de LEETABLA.

b) El sistema envía la información:

 LA MEDIA DE SUS DATOS ES

w termina la ejecución del comando.

##

2.3.4.17 SS/1(FILTRA) .

El mnemónico del comando es S11.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO FILTRA LA SERIE *
 * EJECUTA PRIMERO EL COMANDO LEESERIE *

el comando LEESERIE se ejecuta una sola vez, ya que la transformación se realiza en un conjunto de datos del tipo serie.

b) El sistema envía la instrucción que está localizada físicamente en el archivo SS/FILTRA1.

* OBSERVACION :

ALGORITMO FILTRA1 NO IMPLEMENTADO.
 EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS VALORES DE LA SERIE, QUE ESTA ALMACENADA EN DISCO. TAM SE REFIERE AL TAMAÑO DE LA SERIE POR EJEMPLO: SEAN ARRAY ARRC1:1J,REAL I, DECLARACIONES DE LA RUTINA, ENTONCES LAS INSTRUCCIONES:

```
RESIZE(ARR,TAM)
FOR I:=1 STEP 1 UNTIL TAM DO
  ARRCIJ:=AAACIJ
```

CUMPLIRIAN ESTE OBJETIVO.
 DESPUES DEBE CARGAR LOS NUEVOS VALORES EN CCCCII, PARA ESCRITURA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

el usuario debe contestar NO, hasta que el algoritmo haya sido implementado, y termina el comando.

##

2.3.4.18 SS/2(FILTRA) .

El mnemónico del comando es S12.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO FILTRA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está localizada físicamente en el archivo SS/FILTRA2.

* OBSERVACION :

ALGORITMO FILTRA2 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL LOS VALORES DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

el usuario debe contestar NO, hasta que el algoritmo haya sido implementado, terminando la ejecución del comando.

##

2.3.4.19 SS/3(FILTRA) .

El mnemónico del comando es S13.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO FILTRA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la información que está físicamente localizada en el archivo SS/FILTRA3:

* OBSERVACION :

ALGORITMO FILTRA3 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS VALORES DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que responder NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

##

2.3.4.20 SS/1(ESPECTRO) .

El mnemónico del comando es S21.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE EL ESPECTRO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

Procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está físicamente localizada en el archivo SS/ESPECTRO1 :

* OBSERVACION :

ALGORITMO ESPECTRO1 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS VALORES DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

##

2.3.4.21 SS/2(ESPECTRO) .

El mnemónico del comando es S22.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE EL ESPECTRO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

Procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está físicamente localizada en el archivo SS/ESPECTRO2 :

* OBSERVACION :

ALGORITMO ESPECTRO2 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS VALORES DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

##

2.3.4.22 SS/3(ESPECTRO) .

El mnemónico del comando es S23.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO OBTIENE EL ESPECTRO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

Procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está físicamente localizada en el archivo SS/ESPECTRO3 :

* OBSERVACION :

ALGORITMO ESPECTRO3 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO
QUE EL DEFINA, EL CONTENIDO DEL ARREGLO
AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO
DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS
VALORES DE LA SERIE QUE ESTA ALMACENADA EN
DISCO. *

c) El sistema envia el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

**

2.3.4.23 SS/1(SUAVIZA) .

El mnemónico del comando es S31.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envia los mensajes:

* EL COMANDO SUAVIDA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envia la instrucción que está físicamente localizada en el archivo SS/SUAVIDA1 :

* OBSERVACION :

ALGORITMO SUAVIDA1 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO
QUE EL DEFINA, EL CONTENIDO DEL ARREGLO
AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO
DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS
VALORES DE LA SERIE QUE ESTA ALMACENADA EN
DISCO. *

c) El sistema envia el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

**

2.3.4.24 SS/2(SUAVIDA) .

El mnemónico del comando es S32.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO SUAVIZA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está físicamente localizada en el archivo SS/SUAVIZA2 :

* OBSERVACION :
ALGORITMO SUAVIZA2 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO QUE EL DEFINA, EL CONTENIDO DEL ARREGLO AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS VALORES DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

##

2.3.4.25 SS/3(SUAVIZA) .

El mnemónico del comando es S33.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* EL COMANDO SUAVIZA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la información que está físicamente localizada en el archivo SS/SUAVIZA3 :

* OBSERVACION :
ALGORITMO SUAVIZA3 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO

QUE EL DEFINA, EL CONTENIDO DEL ARREGLO
AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO
DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS
VALORES DE LA SERIE QUE ESTA ALMACENADA EN
DISCO. *

c) El sistema envia el mensaje:

* DESEA ALMACENAR SUS NUEVOS VALORES *

teniendo el usuario que contestar NO, hasta que haya sido implementa-
do el algoritmo, y termina la ejecución del comando.

##

2.3.4.26 SS/(ESCALAR) .

El mnemónico del comando es SSE.

Este comando permite amplificar o simplificar a todos y cada uno
de los elementos de una serie.

Realiza las siguientes acciones:

a) El sistema envia los mensajes:

* EL COMANDO AMPLIFICA O SIMPLIFICA A LOS
ELEMENTOS DE LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) El sistema envia el mensaje:

* ESCRIBA EL ESCALAR POR EL QUE QUIERE MODIFICAR
A TODOS LOS ELEMENTOS DE LA SERIE *

teniendo el usuario que especificar el número real por el que quiere
modificar a la serie.

c) El sistema envia el mensaje:

* DESEA CONOCER SUS RESULTADOS *

si el usuario contesta SI, el sistema envia el mensaje:

*** ESPECIFIQUE EL INTERVALO ***

si el usuario comete algún error, el sistema envia el mensaje :

*** ERROR EN EL INTERVALO ***

EL TAMANO DE LA SERIE VARIA DE 1 A ---

repetiéndose la pregunta sobre la especificación del intervalo.

d) Por último, el sistema envia el mensaje :

* DESEA ALMACENAR LA NUEVA SERIE *

si el usuario contesta SI, se ejecuta el comando ESCRSERIE, de otra manera termina el comando.

##

2.3.4.27 SET/1(IDENT) .

El mnemónico del comando es ST1.

El algoritmo que se usa cuando se ejecuta este comando no está implementado. Sin embargo se simulan ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

- * SE EJECUTA EL COMANDO IDENTIFICA *
- * EL COMANDO TRANSFORMA A LOS ELEMENTOS DE UNA SERIE EN ELEMENTOS DE UNA TABLA. PARA LECTURA EJECUTA LEESERIE, PARA ESCRITURA EJECUTA ESCRTABLE. *

procediéndose a la ejecución de LEESERIE.

b) El sistema envía la instrucción que está físicamente localizada en el archivo SET/IDENTIFICA1 :

* OBSERVACION :

ALGORITMO IDENTIFICA1 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE DEFINIR EN ESTA RUTINA UN ARREGLO EN EL QUE VA A TRANSFERIR LOS DATOS DE LA SERIE QUE ESTAN EN UN ARREGLO DE MEMORIA PRINCIPAL AAACIJ I=1,2,...,TAM (TAM ES EL TAMANO DE LA SERIE), PARA REALIZAR LAS TRANSFORMACIONES PERTINENTES, SI LUEGO DESEA ALMACENAR EN DISCO LOS NUEVOS VALORES OBTENIDOS, LOS DEBE CARGAR COMPLETAMENTE EN EL ARREGLO CCCCII CUYO TAMANO DEBE ESTAR ACTUALIZADO, YA QUE LA RUTINA DE ESCRITURA ESCRTABLE UTILIZA ESTE ARREGLO. POR OTRO LADO EL PROGRAMADOR DEBE PONER MUCHA ATENCION EN LA ORGANIZACION DE LAS TABLAS (SOBRETUDO AL ORDEN N, FILAS M, COLUM P) SI EL NUMERO DE DATOS QUE ESTAN EN CCCCII NO CONCUERDA CON EL PRODUCTO (N+1)*M*P SE PRODUCIRAN ERRORES. *

c) El sistema envía el mensaje:

* DESEA ALMACENAR LOS VALORES OBTENIDOS *

teniendo el usuario que contestar NO, hasta que haya sido implementado el algoritmo, y termina la ejecución del comando.

##

2.3.4.28 SET/2(IDENT) .

El mnemónico del comando es ST2.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en SET/1(IDENT), con la diferencia que la instrucción que envía el sistema está en el archivo físico SET/IDENTIFICA2 :

* OBSERVACION :
ALGORITMO IDENTIFICA2 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE EJECUTAR EL COMANDO
SET/1(IDENT) PARA LAS ACCIONES GENERALES
PERTINENTES EN LA IMPLEMENTACION DE LA
RUTINA. *

##

2.3.4.29 SET/3(IDENT) .

El mnemónico del comando es ST3.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en SET/1(IDENT), con la diferencia de que la instrucción que envía el sistema está en el archivo físico SET/IDENTIFICA3 :

* OBSERVACION :
ALGORITMO IDENTIFICA3 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE EJECUTAR EL COMANDO
SET/1(IDENT) PARA LAS ACCIONES GENERALES
PERTINENTES EN LA IMPLEMENTACION DE LA
RUTINA . *

##

2.3.4.30 SET/1(HISTO) .

El mnemónico del comando es ST4.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en SET/1(IDENT), con la diferencia de que la instrucción que envía el sistema está en el archivo físico SET/HISTOGRAMA1 :

* OBSERVACION :
ALGORITMO HISTOGRAMA1 NO IMPLEMENTADO.
SI EL PROGRAMADOR DESEA MAYOR INFORMACION,
EJECUTE EL COMANDO SET/1(IDENT). *

##

2.3.4.31 SET/2(HISTO) .

El mnemónico del comando es ST5.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en SET/1(IDENT), con la diferencia de que la instrucción que envía el sistema está en el archivo físico SET/HISTOGRAMA2 :

* OBSERVACION :
ALGORITMO HISTOGRAMA2 NO IMPLEMENTADO.
SI EL PROGRAMADOR DESEA MAYOR INFORMACION,
EJECUTE EL COMANDO SET/1(IDENT). *

##

2.3.4.32 SET/3(HISTO) .

El mnemónico del comando es ST6.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en SET/1(IDENT), con la diferencia de que la instrucción que envía el sistema está en el archivo físico SET/HISTOGRAMA3 :

* OBSERVACION :
ALGORITMO HISTOGRAMA3 NO IMPLEMENTADO.
SI EL PROGRAMADOR DESEA MAYOR INFORMACION,
EJECUTE EL COMANDO SET/1(IDENT). *

##

2.3.4.33 TAS/1(SIMULA) .

El mnemónico del comando es TS1.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado, se siguen ciertos pasos que son necesarios para la ejecución del comando.

a) El sistema envía los mensajes:

* SE EJECUTA EL COMANDO SIMULA *
* EL COMANDO SIMULA UNA SERIE A PARTIR
DE LOS ELEMENTOS DE UNA TABLA.
PARA LECTURA EJECUTA EL COMANDO LEETABLA,
PARA ESCRITURA EJECUTA ESCRSERIE. *

Procediéndose a la ejecución de LEETABLA.

b) El sistema envía la instrucción que está físicamente localizada en el archivo TAS/SIMULA1 :

* OBSERVACION :

ALGORITMO SIMULA1 NO IMPLEMENTADO.
LAS INDICACIONES GENERALES PARA IMPLEMENTAR
ESTA RUTINA SON LAS MISMAS QUE SE OBTIENEN
AL EJECUTAR EL COMANDO SET/1(IDENT), CON
LA DIFERENCIA QUE AL MOMENTO DE ESCRITURA
EN DISCO SE EJECUTA EL COMANDO ESCRSERIE.
ESTO YA ESTA PREVISTO POR EL SISTEMA. *

##

2.3.4.34 TAS/2(SIMULA) .

El mnemónico del comando es TS2.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en TAS/1(SIMULA), con la diferencia de que la instrucción que envía el sistema está en el archivo TAS/SIMULA2.

##

2.3.4.35 TAS/3(SIMULA) .

El mnemónico del comando es TS3.

El algoritmo de procesamiento numérico que se usa cuando se ejecuta este comando no está implementado. Se cumplen las indicaciones generales que fueron explicadas en TAS/1(SIMULA), con la diferencia de que la instrucción que envía el sistema está en el archivo físico TAS/SIMULA3.

##

2.3.4.36 GRA/1(CONT) .

El mnemónico del comando es GR1.

Este comando ejecuta un algoritmo de graficación con los datos de una serie.

Realiza las siguientes acciones :

a) El sistema envía los mensajes:

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UNE LOS DIVERSOS PUNTOS EN
FORMA CONTINUA, ES DECIR SE UNEN LOS PUNTOS
P1(X1,Y1),P2(X2,Y2),...,PN(XN,YN). *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

procediéndose a la ejecución de LEESERIE.

b) En el periférico se escribe una w, indicando que el graficador HP del IIMAS está funcionando.

c) Una vez ha realizado el gráfico de la ubicación de los valores de

la serie en el sistema de coordenadas X e Y, donde el eje de las abscisas es el tiempo, y el eje de las ordenadas son los valores $w(i)$ $i=1, 2, \dots, TAM$ de la serie, se marcan los puntos máximo y mínimo de los valores, y termina la ejecución del comando.

##

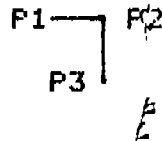
2.3.4.37 GRA/2(HISTO) .

El mnemónico del comando es GR2.

Este comando realiza las mismas acciones que el comando GRA/1(CONT), con la diferencia de que el sistema envía los mensajes:

- * ALGORITMO DE GRAFICACION *
- * OBSERVACION :
- EL ALGORITMO UNE LOS PUNTOS
- P1(X1,Y1),P2(X2,Y1),P3(X2,Y2),P4(X3,Y2),...
- EN FORMA DE HISTOGRAMA. *
- * EJECUTA PRIMERO EL COMANDO LEESERIE *

NOTA: El tipo de HISTOGRAMA que se grafica es HORIZONTAL



##

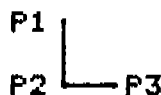
2.3.4.38 GRA/3(HISTO) .

El mnemónico del comando es GR3.

Este comando realiza las mismas acciones que el comando GRA/1(CONT), con la diferencia de que el sistema envía los mensajes:

- * ALGORITMO DE GRAFICACION *
- * OBSERVACION :
- EL ALGORITMO UNE LOS PUNTOS
- P1(X1,Y1),P2(X1,Y2),P3(X2,Y2),P4(X3,Y2),...
- EN FORMA DE HISTOGRAMA*
- * EJECUTA PRIMERO EL COMANDO LEESERIE *

NOTA: El tipo de histograma que se grafica es VERTICAL



##

2.3.4.39 GRA/4(LINEAS) .

El mnemónico del comando es GR4.

Este comando realiza las mismas acciones que el comando GRA/1(CONT), con la diferencia de que el sistema envía los mensajes:

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO DIBUJA LINEAS QUE
PARTIENDO DE LOS PUNTOS $P_1(X_1,0), P_2(X_2,0), \dots$
TERMINAN EN LOS PUNTOS $P_1(X_1,Y_1), P_2(X_2,Y_2), \dots$
RESPECTIVAMENTE. *

##

2.3.4.40 GRA/5(PUNTOS) .

El mnemónico del comando es GR5.

Este comando realiza las mismas acciones que el comando
GRA/1(CONT), con la diferencia de que el sistema envía los mensajes:

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UBICA LOS PUNTOS
 $P_1(X_1,Y_1), P_2(X_2,Y_2), \dots, P_N(X_N,Y_N)$
EN EL PLANO FORMADO POR LAS COORDENADAS X E Y.*

* EJECUTA PRIMERO EL COMANDO LEESERIE *

Como se indicó anteriormente las coordenadas X e Y tienen como
abscisa el tiempo, y como ordenada a los valores $y(i)$ $i=1, 2, \dots,$
TAM de la SERIE.

##

3. PROCEDIMIENTOS DE PROCESAMIENTO NUMERICO .

Como se mencionó anteriormente los diversos procedimientos de
procesamiento numérico, se desarrollan físicamente en forma de árbo-
les independientes entre sí, los cuales a su vez constituyen elemen-
tos de SIMASET.

Los padres o raíces de estos árboles tienen un nombre genérico
ALG/(TIPO DE ALGORITMO), donde tipo de algoritmo puede ser:

- SEN
- SES
- TAT
- TAN
- SS
- SET
- TAS
- GRA

i) Todos los procedimientos del tipo SEN, se refieren a aquellos que
partiendo de un conjunto de números del tipo SERIE, es decir que tie-
nen como entrada los datos de una serie, producen como resultado de
la ejecución un solo número.

SEN ----> SERIE A NUMERO
--- --

Los hijos de ALG/SEN, están identificados con el nombre genérico SEN/(ALGORITMO) donde algoritmo puede ser:

- MAX
- MIN
- MEDIA
- NORMA2
- NORMINF
- NORMA1
- DESVIA

ii) Todos los procedimientos del tipo SES, son aquellos que relacionan a dos series como datos de entrada, y que producen como resultado de la ejecución otra serie que puede ser almacenada en disco.

SES ----> (SERIE, SERIE) A SERIE
--- --

Los hijos de ALG/SES, están identificados con el nombre genérico SES/(ALGORITMO) donde algoritmo puede ser :

- SUMA
- MULT
- CONVOL

iii). Todos los procedimientos del tipo TAT, son aquellos que relacionan a dos tablas como datos de entrada, y que producen como resultado de la ejecución otra tabla, la cual puede ser almacenada en disco.

TAT ----> (TABLA, TABLA) A TABLA
--- --

Los hijos de ALG/TAT, están identificados con el nombre genérico TAT/(ALGORITMO) donde algoritmo puede ser :

- SUMAS
- PREDICTOR
- CONTROLA

iv) Todos los procedimientos del tipo TAN, son aquellos que tienen como entrada los datos de una tabla, y que producen como resultado de la ejecución un solo número.

TAN ----> TABLA A NUMERO
--- --

Los hijos de ALG/TAN, están identificados con el nombre genérico TAN/(ALGORITMO) donde algoritmo puede ser :

- MAXI
- MINI
- MEDI

v) Todos los procedimientos del tipo SS, son aquellos que tienen como entrada los datos de una serie y que producen como resultado de la ejecución otra serie, la cual puede ser almacenada en disco.

SS ---> SERIE A SERIE

--

Los hijos de ALG/SS, están identificados con el nombre genérico SS/(ALGORITMO) donde algoritmo puede ser:

- FILTRA1
- FILTRA2
- FILTRA3
- ESPECTRO1
- ESPECTRO2
- ESPECTRO3
- SUAVIZA1
- SUAVIZA2
- SUAVIZA3
- ESCALAR

vi) Todos los procedimientos del tipo SET, son aquellos que tienen como entrada los datos de una serie y que producen como resultado de la ejecución una tabla que puede ser almacenada en disco.

SET ---> SERIE A TABLA

Los hijos de ALG/SET, están identificados con el nombre genérico SET/(ALGORITMO) donde algoritmo puede ser :

- IDENTIFICA1
- IDENTIFICA2
- IDENTIFICA3
- HISTOGRAMA1
- HISTOGRAMA2
- HISTOGRAMA3

vii) Todos los procedimientos del tipo TAB, se caracterizan porque tienen como entrada los datos de una tabla y producen como resultado de la ejecución una serie, la cual puede ser almacenada en disco.

TAB ---> TABLA A SERIE

Los hijos de ALG/TAB, están identificados con el nombre genérico TAB/(ALGORITMO) donde algoritmo puede ser :

- SIMULA1
- SIMULA2
- SIMULA3

viii) Por último, se tiene al procedimiento de tipo GRA, que incluye a todas las diversas opciones de graficación de las series. Este algoritmo se vincula con la rutina GRAFIK de la biblioteca del HP/GRAF del IIMAS.

III.4. ARCHIVOS DE ALMACENAMIENTO.

Los archivos de almacenamiento de la información están conformados por diversos sub-archivos, y están divididos en dos grupos, dependiendo de la información numérica.

El diseño de los archivos físicos ha sido desarrollado en forma de sub-archivos principalmente debido a que al crear un nuevo archivo independiente para almacenar la información, se genera un descriptor a ese archivo. Este hecho ya fue analizado previamente.

Cada archivo donde se almacena la información propiamente dicha, tiene asociado un directorio a través del cual se realiza la búsqueda de los datos para las operaciones de entrada/salida.

4.1 ARCHIVOS DE SERIES.

Existen dos archivos para el almacenamiento de datos del tipo SERIE:

i) El archivo cuyo nombre lógico es ARCH1, y cuyo nombre físico en disco es "{clave} DE USUARIO"/ARCH1/"Password". tiene registros de una palabra y está formado de diferentes sub-archivos. Cada uno de los sub-archivos tiene un encabezado que consta de:

- número de serie
- nombre particular de la serie
- tamaño de la serie

después de este encabezado se almacenan en disco los datos de la serie.

ii) El archivo cuyo nombre lógico es DIRECTORIO, y cuyo nombre físico en disco es "{clave} DE USUARIO"/DIRECTORIO/"Password". tiene registros de tres palabras, y sirve de directorio al archivo ARCH1. Este archivo tiene un encabezado que consta de:

NOMBRE TAM DFIN

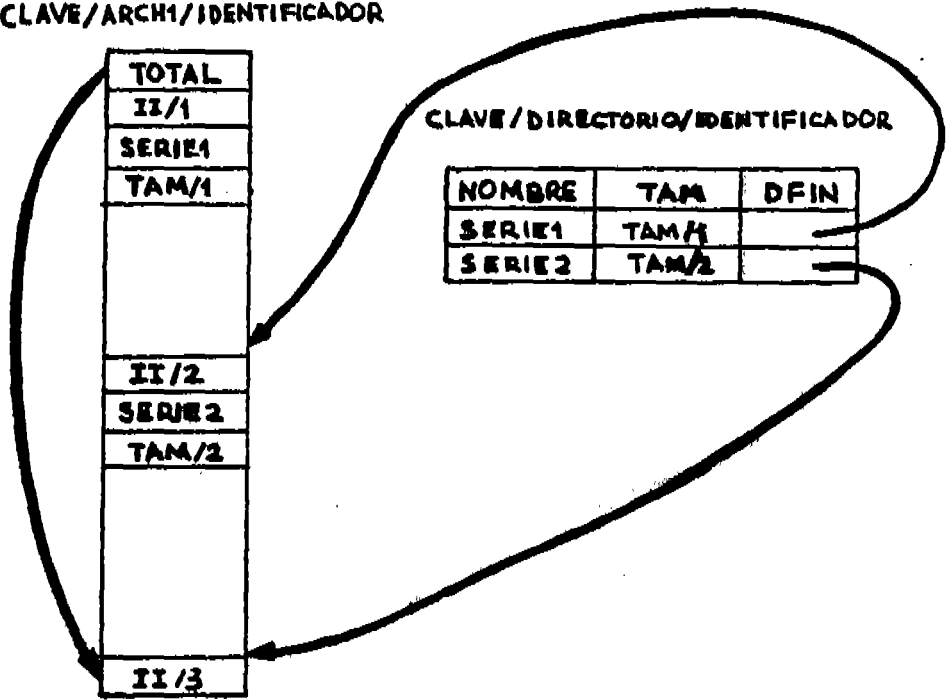
donde NOMBRE se refiere al nombre particular de la serie; TAM se refiere al tamaño, es decir al número de datos que tiene la serie; y DFIN se refiere a la dirección final de la mencionada serie, es decir al último registro del sub-archivo que representa la serie en el archivo ARCH1.

CLAVE/ARCH1/IDENTIFICADOR

TOTAL
II/1
SERIE1
TAM/1
II/2
SERIE2
TAM/2
II/3

CLAVE/DIRECTORIO/IDENTIFICADOR

NOMBRE	TAM	DFIN
SERIE1	TAM/1	---
SERIE2	TAM/2	---



4.2 ARCHIVOS DE TABLAS.

Existen dos archivos para el almacenamiento de datos del tipo TABLA:

i) El archivo cuyo nombre lógico es TABLA, y cuyo nombre físico en disco es "{clave} DE USUARIO"/TABLA/"Password". tiene una configuración similar al archivo de series, es decir está conformado por diferentes sub-archivos, cada uno de los cuales tiene un encabezado que consta de:

número de tabla
 nombre particular de la tabla
 tamaño de la tabla

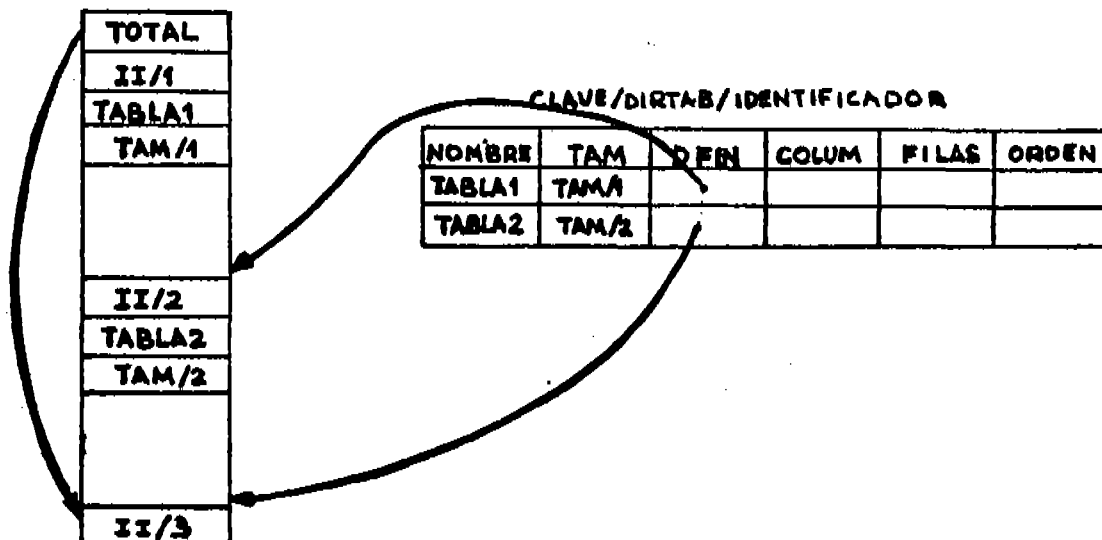
después de este encabezado se almacenan en disco los datos .

ii) El archivo cuyo nombre lógico es DIRTAB y cuyo nombre físico en disco es "{clave} DE USUARIO"/DIRTAB/"Password". tiene registros de seis palabras, y sirve de directorio para el archivo TABLA. Este archivo tiene un encabezado que consta de:

NOMBRE TAM DFIN COLUM FILAS ORDEN

donde NOMBRE es el nombre particular de la tabla; TAM es el tamaño de la tabla; DFIN es la dirección al último registro de la tabla en el archivo TABLA; COLUM es el número de columnas de las matrices que conforman la tabla; FILAS es el número de filas de las matrices de la tabla; y ORDEN es el número de matrices que van a formar la tabla. A este número se le suma uno, ya que de acuerdo a la definición de tabla, existe un número (N+1) de matrices de dimensiones (MxP).

CLAVE/TABLA/IDENTIFICADOR



4.3 DESCRIPCION ESTANDARIZADA DE LOS ARCHIVOS DE SIMASET

Este tipo de descripción, está dirigida principalmente a los usuarios con el propósito fundamental de que tengan alguna forma donde puedan ir agregando y modificando los elementos que se analizan a continuación :

4.3.1 VALORES .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET

FECHA : MARZO/1980

NOMBRE EXTERNO : VALORES

NOMBRE INTERNO : VALORES

MEDIO DE ALMACENAMIENTO : DISCO

TIPO DE CODIGO : ALFANUMERICO (EBCDIC)

LONG. DEL REGISTRO : 14 PALABRAS

FACTOR DE BLOQUEO : 420 REGISTROS

TAMANO MAXIMO DEL ARCHIVO :

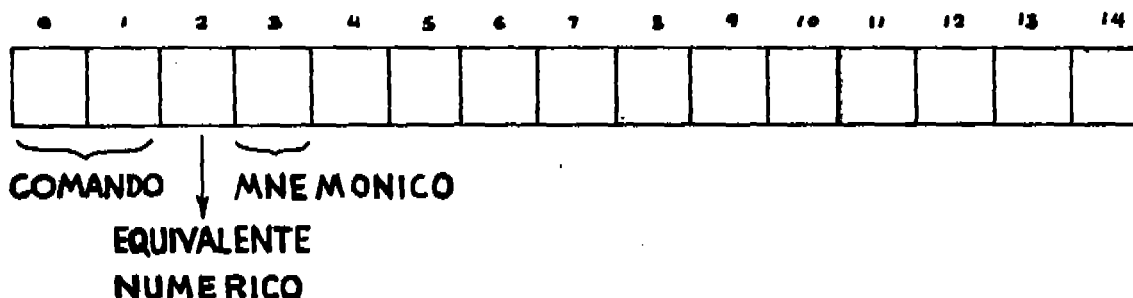
VOLUMEN INICIAL : 53 REGISTROS

CRECIMIENTO ESPERADO :

DESCRIPCION DEL ARCHIVO : ÉSTE ARCHIVO ES CREADO EN DATA Y ALMACENA LOS DIVERSOS COMANDOS DEL SISTEMA CON SU CORRESPONDIENTE EQUIVALENTE NUMERICO DECIMAL, Y LA ABREVIACION DEL COMANDO. ESTE ARCHIVO CONSTITUYE LA SINTESIS DEL NIVEL SINTACTICO DEL SISTEMA.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET

CONFIGURACION DEL REGISTRO :



4.3.2 DIRUSER .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET
FECHA : MARZO/1980
NOMBRE EXTERNO : DIRUSER
NOMBRE INTERNO : DIRUSER
MEDIO DE ALMACENAMIENTO : DISCO
TIPO DE CODIGO : ALFANUMERICO (EBCDIC)
LONG. DEL REGISTRO : 4 PALABRAS
FACTOR DE BLOQUEO : 60 REGISTROS
TAMANO MAXIMO DEL ARCHIVO : 11 REGISTROS
VOLUMEN INICIAL : 11 REGISTROS
CRECIMIENTO ESPERADO : NINGUNO

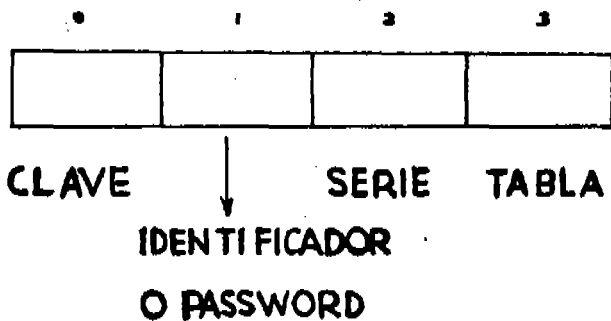
DESCRIPCION DEL ARCHIVO : ESTE ARCHIVO ES GENERADO POR EL PROGRAMA ARCHIVO1. SIRVE DE DIRECTORIO DE USUARIOS. CADA REGISTRO ESTA FORMADO POR 4 CAMPOS O ATRIBUTOS, CADA UNO DE 1 PALABRA DE LONGITUD.

LOS CAMPOS SON :
{clave} DE USUARIO
{id} O PASSWORD
SERIE
TABLA

SI LOS VALORES DE SERIE Y DE TABLA SON 0 SIGNIFICA QUE NO SE HAN CREADO ARCHIVOS FISICOS PARA SERIES O TABLAS.

SI EL VALOR YA SEA DE SERIE O DE TABLA ES 1, SIGNIFICA QUE EXISTEN ARCHIVOS FISICOS PARA SERIES O TABLAS.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET
CONFIGURACION DEL REGISTRO :



4.3.3 ARCH1 .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET

FECHA : MARZO/1980

NOMBRE EXTERNO : ARCH1

NOMBRE INTERNO : "{clave} DE USUARIO"/ARCH1/{id}" .

MEDIO DE ALMACENAMIENTO : DISCO

TIPO DE CODIGO : ALFANUMERICO (EBCDIC)

LONG. DEL REGISTRO : 1 PALABRA

FACTOR DE BLOQUEO : 90 REGISTROS

TAMANO MAXIMO DEL ARCHIVO :

VOLUMEN INICIAL :

CRECIMIENTO ESPERADO :

DESCRIPCION DEL ARCHIVO : ESTE ARCHIVO ES GENERADO POR CADA USUARIO A TRAVES DE SIMASET CON EL COMANDO CREASERIE, Y ES EL ARCHIVO DONDE SE ALMACENAN LOS DIVERSOS SUB-ARCHIVOS QUE CONSTITUYEN LAS SERIES QUE TIENE EL USUARIO. EN EL PRIMER REGISTRO DEL ARCHIVO SE ALMACENA UN APUNTA-
DOR AL ULTIMO REGISTRO ACTUAL DEL ARCHIVO. CADA SUB-ARCHIVO ESTA FORMADO POR EL NUMERO DE LA SERIE, EL NOMBRE DE LA SERIE, EL TAMANO DE LA SERIE Y LOS DATOS DE LA SERIE.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET

CONFIGURACION DEL REGISTRO :



D A T O

4.3.4 DIRECTORIO .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET

FECHA : MARZO/1980

NOMBRE EXTERNO : DIRECTORIO

NOMBRE INTERNO : "{clave} DE USUARIO"/DIRECTORIO/"{id}" .

MEDIO DE ALMACENAMIENTO : DISCO

TIPO DE CODIGO : ALFANUMERICO (EBCDIC)

LONG. DEL REGISTRO : 3 PALABRAS

FACTOR DE BLOQUEO : 30 REGISTROS

TAMANO MAXIMO DEL ARCHIVO :

VOLUMEN INICIAL :

CRECIMIENTO ESPERADO :

DESCRIPCION DEL ARCHIVO : ESTE ARCHIVO ES CREADO POR EL USUARIO A TRAVES DE SIMASET, SIEMPRE Y CUANDO HAYA GENERADO EL ARCHIVO DE SERIES. EL PRIMER REGISTRO DEL ARCHIVO TIENE UN ENCABEZADO:

NOMBRE TAM DFIN

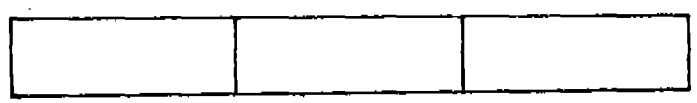
EN ESTOS CAMPOS O ATRIBUTOS SE VAN A ALMACENAR: EL NOMBRE PARTICULAR DE LA SERIE, EL TAMANO DE LA SERIE Y LA DIRECCION FINAL, ES DECIR UN APUNTA- DOR AL ULTIMO REGISTRO DEL SUB-ARCHIVO QUE REPRESENTA LA SERIE EN EL ARCHIVO ARCH1.

ES A TRAVES DE ESTE ARCHIVO QUE SE UBICAN RAPIDAMENTE LAS DIVERSAS SERIES QUE DESEA ESTUDIAR EL USUARIO.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET

CONFIGURACION DEL REGISTRO :

0 . 1 2



NOMBRE TAMAÑO DIRECCION

4.3.5 TABLA .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET

FECHA : MARZO/1980

NOMBRE EXTERNO : TABLA

NOMBRE INTERNO : "{clave} DE USUARIO"/TABLA/"{id}"

MEDIO DE ALMACENAMIENTO : DISCO

TIPO DE CODIGO : ALFANUMERICO (EBCDIC)

LONG. DEL REGISTRO : 1 PALABRA

FACTOR DE BLOQUEO : 30 REGISTROS

TAMANO MAXIMO DEL ARCHIVO :

VOLUMEN INICIAL :

CRECIMIENTO ESPERADO :

DESCRIPCION DEL ARCHIVO : ESTE ARCHIVO ES GENERADO POR EL USUARIO A TRAVES DE SIMASET, CON EL COMANDO CREATABLA Y ES EL ARCHIVO DONDE SE ALMACENAN LOS DIVERSOS SUB-ARCHIVOS QUE CONSTITUYEN LAS TABLAS QUE TIENE EL USUARIO. EN EL PRIMER REGISTRO SE ALMACENA UN APUNTADOR AL ULTIMO REGISTRO ACTUAL DEL ARCHIVO. CADA SUB-ARCHIVO ESTA FORMADO POR:
EL NUMERO DE LA TABLA, EL NOMBRE DE LA TABLA, EL TAMANO DE LA TABLA Y LOS DATOS DE LA TABLA.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET

CONFIGURACION DEL REGISTRO :



DATO

4.3.6 DIRTAB .

DESCRIPCION DEL ARCHIVO .

SISTEMA : SIMASET

FECHA : MARZO/1980

NOMBRE EXTERNO : DIRTAB

NOMBRE INTERNO : "{clave} DE USUARIO"/DIRTAB/"{id}"

MEDIO DE ALMACENAMIENTO : DISCO

TIPO DE CODIGO : ALFANUMERICO (EBCDIC)

LONG. DEL REGISTRO : 6 PALABRAS

FACTOR DE BLOQUEO : 30 REGISTROS

TAMANO MAXIMO DEL ARCHIVO :

VOLUMEN INICIAL :

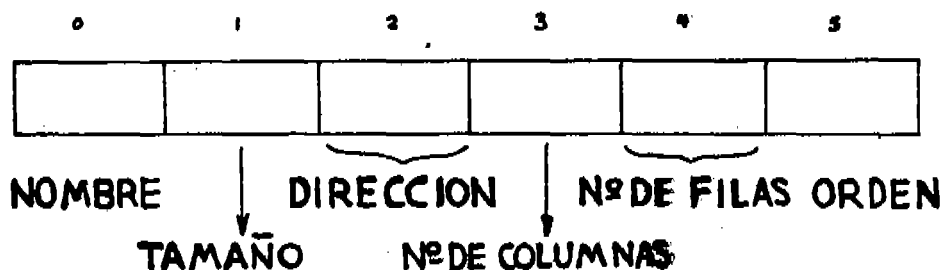
CRECIMIENTO ESPERADO :

DESCRIPCION DEL ARCHIVO : SI EL USUARIO GENERO SU ARCHIVO DE TABLAS, SE CREA ESTE ARCHIVO QUE SIRVE DE DIRECTORIO PARA LAS TABLAS EL PRIMER REGISTRO DEL ARCHIVO TIENE EL ENCABEZADO:

NOMBRE TAM DFIN COLUM FILAS ORDEN
EN ESTOS CAMPOS SE ALMACENAN: EL NOMBRE PARTICULAR DE LA TABLA, EL TAMANO DE LA TABLA, LA DIRECCION FINAL AL ULTIMO REGISTRO DE LA TABLA, EL NUMERO DE COLUMNAS DE LAS MATRICES, EL NUMERO DE FILAS DE LAS MATRICES, Y EL NUMERO DE MATRICES QUE TIENE LA TABLA.

PROGRAMA QUE UTILIZA EL ARCHIVO : SIMASET

CONFIGURACION DEL REGISTRO :



III.5. OPERACIONES PARA EL INGRESO DE NUEVOS COMANDOS A SIMASET .

Por último para terminar la descripción de SIMASET, se considera importante el indicar las diversas acciones que se deben realizar para el caso de aumentar comandos o instrucciones al sistema y sus correspondientes nuevos procedimientos de procesamiento numérico, que van a estar localizados en los padres de las familias de los diversos módulos planteados.

1) Al comienzo del nuevo procedimiento se deben declarar las instrucciones:

```
* SET LEVEL 3
* SET AUTOBIND
* HOST IS OBJECT/SIMASET
[
*LOADINFO "ISAL/SIMASET/INFO"
]
```

la acción de LOADINFO, es de pasar variables globales del sistema al nuevo procedimiento.

Se implementa el procedimiento, se lo compila como SIMASET y guarda.

NOTA: Si el padre de la familia de los procedimientos de este tipo usa lectura, los datos están en el arreglo AAC[I] I=1, 2, ...TAM. Si se quieren conocer los resultados o almacenarlos en disco, debe usarse el arreglo CCC[I] I=1, 2, ...TAM.

2) Se debe en la parte inicial del padre de la familia de los procedimientos respectiva, declarar al nuevo procedimiento como EXTERNAL, usarlo en el lugar correspondiente del mecanismo de búsqueda de éste, compilar al padre como SIMASET y guardarlo.

3) Se debe crear y agregar el nuevo comando en el archivo VALORES. Este tiene registros de 4 palabras, en el que las columnas de 0-14 se usan para el comando explícito, las columnas de 15-17 se usan para el correspondiente número o equivalente numérico del comando, y las columnas de la 21-23 se usan para el mnemónico del comando. Se debe guardar el archivo actualizado.

4) Por último se debe hacer el nuevo llamado del padre de la familia de procedimientos respectiva en el programa PRIN/EJECUTA. Así mismo se debe actualizar la variable NNN que se encuentra en la línea 2140 de PRIN/EJECUTA. Si se agrega solo un comando, ésta debe aumentarse en una unidad. En la actualidad su valor es NNN=52. Una vez realizados estos cambios se compila como SIMASET, y se guarda, estando el sistema nuevamente listo para ejecución.

IV. FUNCIONAMIENTO Y RESULTADOS

Es en la práctica donde se deben verificar y solidificar las diversas teorías.

Como se explicó en el capítulo III de la descripción física de SIMASET, el sistema consta de un conjunto de programas de diverso tipo. Los listados de éstos se incluyen en el APENDICE A.

Así mismo se considera muy importante, el incluir diversos ejemplos y resultados del funcionamiento de SIMASET.

IV.1. INGRESO AL SISTEMA .

Una vez el usuario ha iniciado una sesión en la B 6800, debe escribir por el periférico :

R SIMASET

El sistema envía los mensajes:

*** CUAL ES SU CLAVE DE USUARIO ***

El usuario debe especificar los 4 caracteres de su clave.

*** CUAL ES SU IDENTIFICADOR O PASSWORD ***

El usuario debe indicar el número que identifica completamente a la clave, y que le permite o no ingresar al sistema.

En el caso de que ya existan 10 usuarios, el sistema envía el mensaje:

{* NO SE LE PUEDE ADMITIR AL SISTEMA *}

En el caso de que el usuario comete algún error en el password, el sistema envía el mensaje:

{* ERROR EN SU CLAVE DE USUARIO *}

terminando en los 2 casos la ejecución de SIMASET.

Si no existe ningún contratiempo en esta operación, el sistema envía la información :

*** BIENVENIDO AL SISTEMA SIMASET ***
*** COMANDOS ***

lo que significa que el usuario puede comenzar una sesión de trabajo. En el caso de que comete algún error en el comando, se obtiene por el

periférico el mensaje:

*** ERROR SINTACTICO ***
**

El usuario tiene la opción de escribir por el periférico ya sea el comando explícito, o el mnemónico de dicho comando.

Una vez el usuario ha realizado todas las operaciones que actúan ya sea sobre SERIES o sobre TABLAS, debe escribir el comando:

FIN

terminando la ejecución de SIMASET.

A más de esto el usuario puede hacer uso de la B 6800, y listar directamente los archivos que son los directorios de SERIES y de TABLAS.

Por ejemplo. Si la clave del usuario es MDHP, y su identificador es 1, puede escribir:

L MDHP/DIRECTORIO/1
L MDHP/DIRTAB/1

obteniendo el listado de todos los registros de estos archivos, que son los directorios de SERIES y de TABLAS.

No se aconseja que se listen desde CANDE los archivos de SERIES o de TABLAS propiamente dichos, ya que estos contienen información no decifrabable por el usuario.

Para terminar la sesión en la B 6800 el usuario debe teclear:

BYE

obteniendo la información:

#END SESSION --- ET=--- PT=--- IO=---
#USER = "{clave}" "LA HORA Y LA FECHA"

IV.2. RESULTADOS Y EJEMPLOS .

Para organizar en mejor forma esta sección, se ejecutan los comandos en grupos, dependiendo de su operación, ya sea de SERVICIO GENERAL, PROCESAMIENTO NUMERICO, y GRAFICACION.

INGRESO AL SISTEMA.

```
R SIMASET
#RUNNING 1609
*** CUAL ES SU CLAVE DE USUARIO ***
#?
LFHP
*** CUAL ES SU IDENTIFICADOR O PASSWORD ***
23

*** BIENVENIDO AL SISTEMA SIMASET ***
***COMANDOS***
```

2.1 EJECUCION DE COMANDOS DE SERVICIO GENERAL.

CREASERIE

```
* LOS ARCHIVOS FISICOS PARA SERIES SON :
  LFHP/ARCH1/23.
  LFHP/DIRECTORIO/23.
```

##

CREATABLA

```
* LOS ARCHIVOS FISICOS PARA TABLAS SON :
  LFHP/TABLA/23.
  LFHP/DIRTAB/23
```

##

ESCRSERIE

{* DE UN MAXIMO DE 6 CARACTERES PARA
EL NOMBRE DEL ARCHIVO A CREAR *)

SERIE1

{* ESCRIBA SUS DATOS *)

23.5

12

21

24.25

30.89

5

6

7

8.9

22.25

56

98

67

56

45

34

?END

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

#?

SI

{* EL TAMANO DE LA SERIE ES= 16 *)

{* EN CASO DE HABER ESCRITO EN DISCO

{* DESEA UN LISTADO?CONTESTE SI O NO *)

NO

##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}
SERIE1
{*DESEA CONOCER EL TAMANO DE LA SERIE*}
SI
{* EL TAMANO DE LA SERIE ES = 16 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}
SI

{* ESPECIFIQUE EL INTERVALO *}
1,16

{* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *}

DATO [1]	23.50000
DATO [2]	12.00000
DATO [3]	21.00000
DATO [4]	24.25000
DATO [5]	30.89000
DATO [6]	5.00000
DATO [7]	6.00000
DATO [8]	7.00000
DATO [9]	8.90000
DATO [10]	22.25000
DATO [11]	56.00000
DATO [12]	98.00000
DATO [13]	67.00000
DATO [14]	56.00000
DATO [15]	45.00000
DATO [16]	34.00000

##

ACTUSERIE

{* CUAL ARCHIVO VA A ACTUALIZAR *}
SERIE1
{* ESCRIBA: POSICION,DATO,NUEVODATO *}
3,21,21.7589
{* DESEA UN LISTADO?CONTESTE SI O NO *}
SI

{* ESPECIFIQUE EL INTERVALO *}
2,4

{* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *}

DATO [2]	12.00000
DATO [3]	21.75890
DATO [4]	24.25000

##

NOTA : Se leen las series : DATOS3 y DATOS4, y se listan todos sus elementos con el propósito de comparar los resultados cuando se ejecuta el comando SES/1(SUMA) .

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}
DATOS3
{*DESEA CONOCER EL TAMANO DE LA SERIE*}
SI
{* EL TAMANO DE LA SERIE ES = 23 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}
SI
{* ESPECIFIQUE EL INTERVALO *}

1,23

{* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *}

DATO [1]	90.26000
DATO [2]	12.34000
DATO [3]	89.70000
DATO [4]	45.45000
DATO [5]	100.12000
DATO [6]	89.70000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	8.00000
DATO [11]	9.00000
DATO [12]	45.78000
DATO [13]	54.25000
DATO [14]	70.70000
DATO [15]	-67.80000
DATO [16]	8.00000
DATO [17]	9.00000
DATO [18]	886.55000
DATO [19]	445.89800
DATO [20]	-446.85500
DATO [21]	4.00000
DATO [22]	56.00000
DATO [23]	9.00000

##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}

DATOS4

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

SI

{* EL TAMANO DE LA SERIE ES = 21 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

1,21

{* A CONTINUACION SE LISTAN LOS

DATOS ORIGINALES DE DISCO *}

DATO [1]	34.89000
DATO [2]	-78.98760
DATO [3]	56.00000
DATO [4]	56.00000
DATO [5]	34.00000
DATO [6]	12.00000
DATO [7]	3.00000
DATO [8]	4.00000
DATO [9]	6.00000
DATO [10]	7.00000
DATO [11]	8.00000
DATO [12]	7.00000
DATO [13]	6.00000
DATO [14]	5.00000
DATO [15]	4.00000
DATO [16]	8.00000
DATO [17]	9.00000
DATO [18]	67.00000
DATO [19]	-789.00000
DATO [20]	56.00000
DATO [21]	45.00000

##

NOTA : Algunos comandos de procesamiento numérico se ejecutarán sobre el conjunto de datos del tipo tabla : TABLAB

LEETABLA

{* QUE ARCHIVO QUIERE LEER *}

TABLAB

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

SI

{* EL ORDEN N DE LA TABLA ES = 2 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO EN N *}

0,1

{* A CONTINUACION SE LISTAN LOS DATOS ORIGINALES DE DISCO *}

LOS COEFICIENTES DE N = 0 SON

R1.

1	114.0000	2	113.0000	3	112.0000
4	111.0000	5	110.0000	6	109.0000
7	108.0000	8	107.0000	9	1.0000
10	2.0000	11	23.0000	12	7.0000
13	5.0000	14	89.8989	15	5.0000
16	4.0000	17	3.0000		

R2.

1	6.0000	2	7.0000	3	8.0000
4	9.0000	5	8.0000	6	6.0000
7	67.0000	8	-9.6700	9	56.8950
10	45.0000	11	3.0000	12	232.0000
13	898.0000	14	23.0000	15	7.0000
16	8.0000	17	78.0000		

LOS COEFICIENTES DE N = 1 SON

R1.

1	35.0000	2	36.0000	3	7.0000
4	0.0000	5	39.0000	6	40.0000
7	41.0000	8	0.0000	9	43.0000
10	44.0000	11	45.0000	12	46.0000
13	47.0000	14	48.0000	15	49.0000
16	50.0000	17	51.0000		

R2.

1	52.0000	2	53.0000	3	54.0000
4	78.9800	5	789.8900	6	45.0000
7	21.0000	8	-8987.0000	9	543543.0000
10	4.0000	11	34.0000	12	45.0000
13	5.0000	14	5.0000	15	8978.8900
16	5.0000	17	5.0000		

##

QUITASERIE

{* CUAL ARCHIVO DESEA SUPRIMIR *}
SERIE1
* SE ELIMINA EL ARCHIVO *
SERIE1
##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}
SERIE1
{* NO EXISTE ESE ARCHIVO EN DISCO *}
##

Es justamente la serie que se acaba de eliminar del archivo de SERIES.

LIT

{* DEBIO HACER OPERACION DE I/O *}
##

CES

{* YA CREGO SU ARCHIVO DE SERIES *}
##

TERMINA

{* SE ELIMINA DE LA LISTA DE USUARIOS*}
{* SE ELIMINAN SUS ARCHIVOS DE SERIES*}
{* SE ELIMINAN SUS ARCHIVOS DE TABLAS*}
##

2.2 EJECUCION DE COMANDOS DE PROCESAMIENTO NUMERICO .-

NOTA : Varios comandos de procesamiento numérico se ejecutan con referencia al conjunto de datos del tipo serie DATOS1. Así mismo todos los comandos de graficación actúan sobre esta serie.

SEN/1(MAX)

* EL COMANDO OBTIENE EL MAXIMO VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

SI
{* EL TAMANO DE LA SERIE ES = 22 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI
{* ESPECIFIQUE EL INTERVALO *}

1,22

{* A CONTINUACION SE LISTAN LOS DATOS ORIGINALES DE DISCO *}

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

EL VALOR MAXIMO ES
DATO [4] 100.00000

##

SEN/2(MIN)

* EL COMANDO OBTIENE EL MINIMO
VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1
{*DESEA CONOCER EL TAMANO DE LA SERIE*}
NO
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO

EL VALOR MINIMO ES
DATO [10] -70.00000

##

SN3

* EL COMANDO OBTIENE LA MEDIA DE LOS
DATOS ALMACENADOS EN DISCO *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1
{*DESEA CONOCER EL TAMANO DE LA SERIE*}
NO
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO

LA MEDIA DE SUS DATOS ES
24.89000

##

SN4

* EL COMANDO OBTIENE LA NORMA 2 *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

SI

{* EL TAMANO DE LA SERIE ES = 22 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

LA NORMA 2 DE LA SERIE
ES 199.08130

##

SN6

* EL COMANDO OBTIENE LA NORMA 1 *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

8,11

{* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *}

DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000

LA NORMA 1 DE LA SERIE
ES 687.58000

##

TN1

* EL COMANDO OBTIENE EL MAXIMO
VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEETABLA *
{* QUE ARCHIVO QUIERE LEER *}

TABLAS

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

N=2.0, F=17.0, M=2.0,

EL VALOR MAXIMO ESTA EN LA MATRIZ :

ORDEN N	1
REGLON M	2
COLUMNA P	9
Y ES =	543543.00000

##

TN2

* EL COMANDO OBTIENE EL MINIMO
VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEETABLA *
{* QUE ARCHIVO QUIERE LEER *}

TABLAS

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

N=2.0, P=17.0, M=2.0,

EL VALOR MINIMO ESTA EN LA MATRIZ :

ORDEN N	1
REGLON M	2
COLUMNA P	8
Y ES =	-8987.00000

##

SES/1(SUMA)

* EL COMANDO SUMA LOS VALORES DE
LAS SERIES ALMACENADAS EN DISCO *
* EJECUTA DOS VECES EL COMANDO
LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS4

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

{* QUE ARCHIVO QUIERE LEER *}

DATOS3

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

* DESEA ALMACENAR SUS NUEVOS VALORES *

SI

{* DE UN MAXIMO DE 6 CARACTERES PARA
EL NOMBRE DEL ARCHIVO A CREAR *}

SERIE1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

SI

{* EL TAMANO DE LA SERIE ES= 23 *}

{* EN CASO DE HABER ESCRITO EN DISCO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

1,23

{* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *}

DATO [1]	125.15000
DATO [2]	-66.64760
DATO [3]	145.70000
DATO [4]	101.45000
DATO [5]	134.12000
DATO [6]	101.70000
DATO [7]	8.00000
DATO [8]	10.00000
DATO [9]	13.00000
DATO [10]	15.00000
DATO [11]	17.00000
DATO [12]	52.78000
DATO [13]	60.25000
DATO [14]	75.70000
DATO [15]	-63.80000
DATO [16]	16.00000
DATO [17]	18.00000
DATO [18]	953.55000
DATO [19]	-343.10200
DATO [20]	-390.85500
DATO [21]	49.00000
DATO [22]	56.00000
DATO [23]	9.00000

**

NOTA : Se escriben las siguientes tablas en disco, con el propósito de ejemplificar al comando TAT/2(PREDICT) .

ETA

```
{* DE UN MAXIMO DE 6 CARACTERES PARA
    EL NOMBRE DEL ARCHIVO A CREAR *}
TABL1
  {* CUAL ES EL ORDEN DE LA TABLA *}
5
  {* DEFINA EL NUMERO DE COLUMNAS *}
1
  {* DEFINA EL NUMERO DE FILAS *}
1
    {* ESCRIBA SUS DATOS *}
1
2
3
4
5
6
{* EN CASO DE HABER ESCRITO EN DISCO
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO
##
```

ETA

```
{* DE UN MAXIMO DE 6 CARACTERES PARA
    EL NOMBRE DEL ARCHIVO A CREAR *}
TABL2
  {* CUAL ES EL ORDEN DE LA TABLA *}
5
  {* DEFINA EL NUMERO DE COLUMNAS *}
1
  {* DEFINA EL NUMERO DE FILAS *}
1
    {* ESCRIBA SUS DATOS *}
9
8
7
6
5
4
{* EN CASO DE HABER ESCRITO EN DISCO
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO
##
```

TAT/2(PREDICT)

* EL COMANDO OBTIENE LOS COEFICIENTES FS Y US DEL
PREDICTOR. *

* EJECUTA EL COMANDO LEETABLA *
{* QUE ARCHIVO QUIERE LEER *}

TABL1
{* DESEA CONOCER EL ORDEN DE LA TABLA*}

SI
{* EL ORDEN N DE LA TABLA ES = 5 *}
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO

{* QUE ARCHIVO QUIERE LEER *}

TABL2
{* DESEA CONOCER EL ORDEN DE LA TABLA*}

NO
{* DESEA UN LISTADO?CONTESTE SI O NO *}
NO

* CUAL ES EL PASO K DEL PREDICTOR *

3
* DESEA CONOCER SUS RESULTADOS *
SI

LOS VALORES DE F SON

F [1]	8.00000
F [2]	-2.00000

LOS VALORES DE U SON

U [0]	-10.00000
U [1]	-18.00000
U [2]	-26.00000
U [3]	-34.00000
U [4]	-38.00000
U [5]	12.00000

* DESEA ALMACENAR SUS NUEVOS VALORES *
SI

{* OBSERVACION:
EL ORDEN DE SU NUEVA TABLA ES 1
EL NUMERO DE COLUMNAS ES 6
EL NUMERO DE FILAS ES 1 *}

{* DE UN MAXIMO DE 6 CARACTERES PARA
EL NOMBRE DEL ARCHIVO A CREAR *}

TABL3
{* CUAL ES EL ORDEN DE LA TABLA *}

1
{* DEFINA EL NUMERO DE COLUMNAS *}

6
{* DEFINA EL NUMERO DE FILAS *}

1
{* EN CASO DE HABER ESCRITO EN DISCO
{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO
##

LAT

{* QUE ARCHIVO QUIERE LEER *}

TABL3

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

SI

{* EL ORDEN N DE LA TABLA ES = 1 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO EN N *}

0,1

{* A CONTINUACION SE LISTAN LOS DATOS ORIGINALES DE DISCO *}

LOS COEFICIENTES DE N = 0 SON

R1.

1	8.0000	2	-2.0000	3	0.0000
4	0.0000	5	0.0000	6	0.0000

LOS COEFICIENTES DE N = 1 SON

R1.

1	-10.0000	2	-18.0000	3	-26.0000
4	-34.0000	5	-38.0000	6	12.0000

##

TN1

* EL COMANDO OBTIENE EL MAXIMO VALOR GLOBAL DE SUS DATOS *

* EJECUTA PRIMERO EL COMANDO LEETABLA *

{* QUE ARCHIVO QUIERE LEER *}

TABL3

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

N=1.0, P=6.0, M=1.0,

EL VALOR MAXIMO ESTA EN LA MATRIZ :

ORDEN N	1	
RENGLON M	1	
COLUMNA P	6	
Y ES =		12.00000

##

TN2

```
* EL COMANDO OBTIENE EL MINIMO
  VALOR GLOBAL DE SUS DATOS *
* EJECUTA PRIMERO EL COMANDO LEETABLA *
  { * QUE ARCHIVO QUIERE LEER * }
```

TABL3

```
{ * DESEA CONOCER EL ORDEN DE LA TABLA * }
NO
{ * DESEA UN LISTADO? CONTESTE SI O NO * }
NO
N=1.0, F=6.0, M=1.0,
```

EL VALOR MINIMO ESTA EN LA MATRIZ :

```
ORDEN N      1
REGLON M     1
COLUMNA P    5
Y ES =      -38.00000
```

##

TN3

```
* EL COMANDO OBTIENE LA MEDIA DE LOS
  DATOS ALMACENADOS EN DISCO *
* EJECUTA PRIMERO EL COMANDO LEETABLA *
  { * QUE ARCHIVO QUIERE LEER * }
```

TABL3

```
{ * DESEA CONOCER EL ORDEN DE LA TABLA * }
NO
{ * DESEA UN LISTADO? CONTESTE SI O NO * }
NO
```

LA MEDIA DE SUS DATOS ES

-9.00000

##

SS/(ESCALAR)

* EL COMANDO AMPLIFICA O SIMPLIFICA A LOS
ELEMENTOS DE LA SERIE. *

* EJECUTA PRIMERO EL COMANDO LEESERIE *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

SI

{* EL TAMANO DE LA SERIE ES = 22 *}

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

* ESCRIBA EL ESCALAR POR EL QUE QUIERE MODIFICAR
A TODOS LOS ELEMENTOS DE LA SERIE. *

50

* DESEA CONOCER SUS RESULTADOS *

SI

* ESPECIFIQUE EL INTERVALO *

1,22

DATO [1]	3617.50000
DATO [2]	1700.00000
DATO [3]	3399.00000
DATO [4]	5000.00000
DATO [5]	2862.50000
DATO [6]	200.00000
DATO [7]	250.00000
DATO [8]	300.00000
DATO [9]	350.00000
DATO [10]	-3500.00000
DATO [11]	450.00000
DATO [12]	400.00000
DATO [13]	350.00000
DATO [14]	250.00000
DATO [15]	2800.00000
DATO [16]	2250.00000
DATO [17]	1100.00000
DATO [18]	1700.00000
DATO [19]	2800.00000
DATO [20]	350.00000
DATO [21]	400.00000
DATO [22]	350.00000

* DESEA ALMACENAR LA NUEVA SERIE *

NO

##

SS/1(FILTRA)

* EL COMANDO FILTRA LA SERIE *
* EJECUTA PRIMERO EL COMANDO LEESERIE *
 { * QUE ARCHIVO QUIERE LEER * }

DATOS1

{ * DESEA CONOCER EL TAMANO DE LA SERIE * }

NO

{ * DESEA UN LISTADO? CONTESTE SI O NO * }

NO

* OBSERVACION:

ALGORITMO FILTRA1 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO
QUE EL DEFINA, EL CONTENIDO DEL ARREGLO
AAACIJ I=1,2,...,TAM, QUE ES EL ARREGLO
DONDE SE CARGA EN MEMORIA PRINCIPAL, LOS
VALORES DE LA SERIE QUE ESTA ALMACENADA EN
DISCO. TAM SE REFIERE AL TAMANO DE LA SERIE
POR EJEMPLO: SEAN ARRAY ARRCI:1J; REAL I;
DECLARACIONES DE LA RUTINA, ENTONCES LAS
INSTRUCCIONES

RESIZE(ARR,TAM);

FOR I:=1 STEP 1 UNTIL TAM DO

ARRCIJ:=AAACIJ;

CUMPLIRIAN ESTE OBJETIVO.

DESPUES DEBE CARGAR LOS NUEVOS VALORES EN
CCCCIJ PARA ESCRITURA EN DISCO. *

* DESEA ALMACENAR LA NUEVA SERIE *

NO

##

SET/1(IDENT)

* SE EJECUTA EL COMANDO IDENTIFICA *
* EL COMANDO TRANSFORMA A LOS ELEMENTOS
DE UNA SERIE EN ELEMENTOS DE UNA TABLA
PARA LECTURA EJECUTA LEESERIE,
PARA ESCRITURA EJECUTA ESCRtabla. *
{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

* OBSERVACION :

ALGORITMO IDENTIFICA1 NO IMPLEMENTADO.
EL PROGRAMADOR DEBE DEFINIR EN ESTA RUTINA
UN ARREGLO EN EL QUE VA A TRANSFERIR LOS
DATOS DE LA SERIE QUE ESTAN EN UN ARREGLO
DE MEMORIA PRINCIPAL AAACIJ I=1,2,...,TAM
(TAM ES EL TAMANO DE LA SERIE), PARA REA-
LIZAR LAS TRANSFORMACIONES PERTINENTES.
SI LUEGO DESEA ALMACENAR EN DISCO LOS NUE-
VOS VALORES OBTENIDOS, LOS DEBE CARGAR COMPLE-
TAMENTE AL ARREGLO CCCCII, CUYO TAMANO DE-
BE ESTAR ACTUALIZADO, YA QUE LA RUTINA DE
ESCRITURA ESCRtabla UTILIZA ESTE ARREGLO.
POR OTRO LADO EL PROGRAMADOR DEBE PONER MU-
CHA ATENCION EN LA ORGANIZACION DE LAS TA-
BLAS (SOBRETOD0 AL ORDEN N,FILAS M,COLUM P)
SI EL NUMERO DE DATOS QUE ESTAN EN CCCCII
NO CONCUERDA CON EL PRODUCTO (N+1)*M*P SE
PRODUCIRAN ERRORES. *

* DESEA ALMACENAR LOS VALORES OBTENIDOS *

NO

##

TAS/1(SIMULA)

* SE EJECUTA EL COMANDO SIMULA *
* EL COMANDO SIMULA UNA SERIE A PARTIR
DE LOS ELEMENTOS DE UNA TABLA.
PARA LECTURA EJECUTA EL COMANDO LEETABLA
PARA ESCRITURA EJECUTA ESCRSERIE. *
{* QUE ARCHIVO QUIERE LEER *}

TABL1

{* DESEA CONOCER EL ORDEN DE LA TABLA*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

* OBSERVACION :

ALGORITMO SIMULA1 NO IMPLEMENTADO.
LAS INDICACIONES GENERALES PARA IMPLEMENTAR
ESTA RUTINA SON LAS MISMAS QUE SE OBTIENEN
AL EJECUTAR EL COMANDO SET/1(IDENT) , CON
LA DIFERENCIA QUE AL MOMENTO DE ESCRITURA
EN DISCO SE EJECUTA EL COMANDO ESCRSERIE.
ESTO YA ESTA PREVISTO POR EL SISTEMA. *

* DESEA ALMACENAR LOS VALORES OBTENIDOS *

NO

##

2.3 EJECUCION DE COMANDOS DE GRAFICACION.

GR1

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UNE LOS DIVERSOS PUNTOS EN
FORMA CONTINUA, ES DECIR SE UNEN LOS PUNTOS
P1(X1,Y1),P2(X2,Y2),...,PN(XN,YN). *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

w

##

LEESERIE

(* QUE ARCHIVO QUIERE LEER *)

DATOS1

(*DESEA CONOCER EL TAMANO DE LA SERIE*)

SI
(* EL TAMANO DE LA SERIE ES = 22 *)

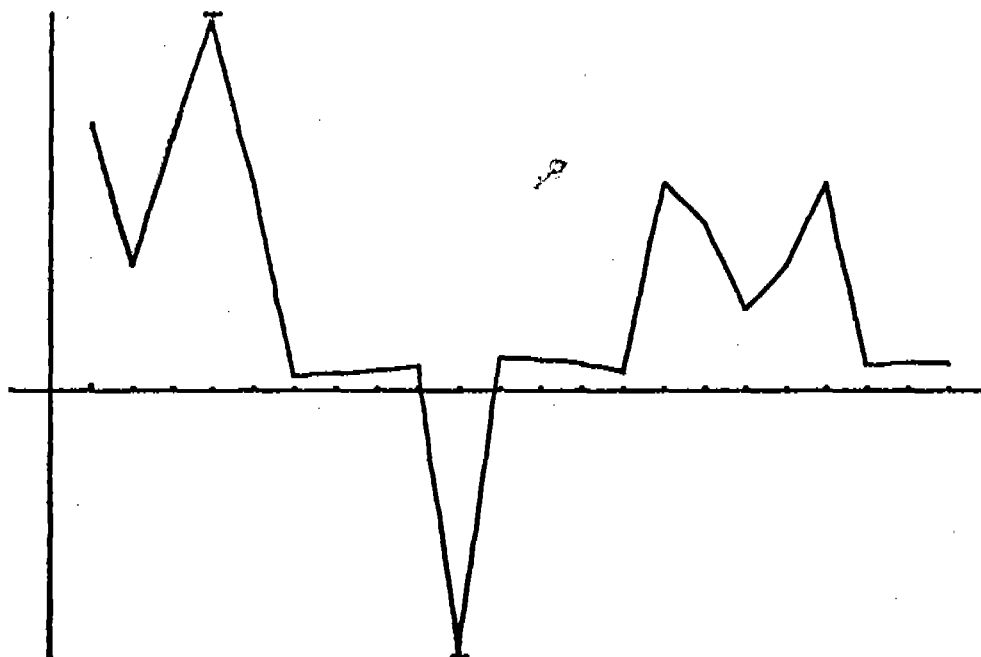
(* DESEA UN LISTADO?CONTESTE SI O NO *)

SI
(* ESPECIFIQUE EL INTERVALO *)

1,22
(* A CONTINUACION SE LISTAN LOS
DATOS ORIGINALES DE DISCO *)

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

##



GR2

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UNE LOS PUNTOS

P1(X1,Y1),P2(X2,Y1),P3(X2,Y2),P4(X3,Y2),...

EN FORMA DE HISTOGRAMA. *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

w

##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

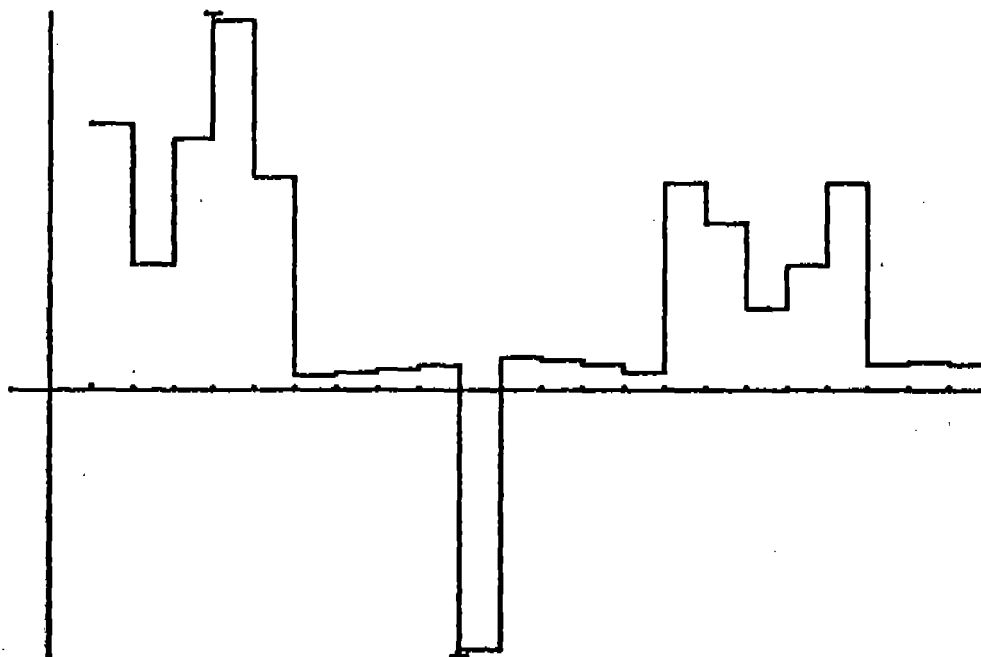
1,22

{* A CONTINUACION SE LISTAN LOS

DATOS ORIGINALES DE DISCO *}

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

##



GR3

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UNE LOS PUNTOS

P1(X1,Y1),P2(X1,Y2),P3(X2,Y2),P4(X2,Y3),...

EN FORMA DE HISTOGRAMA. *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

W

##

LEESERIE

(* QUE ARCHIVO QUIERE LEER *)

DATOS1

(*DESEA CONOCER EL TAMANO DE LA SERIE*)

NO

(* DESEA UN LISTADO?CONTESTE SI O NO *)

SI

(* ESPECIFIQUE EL INTERVALO *)

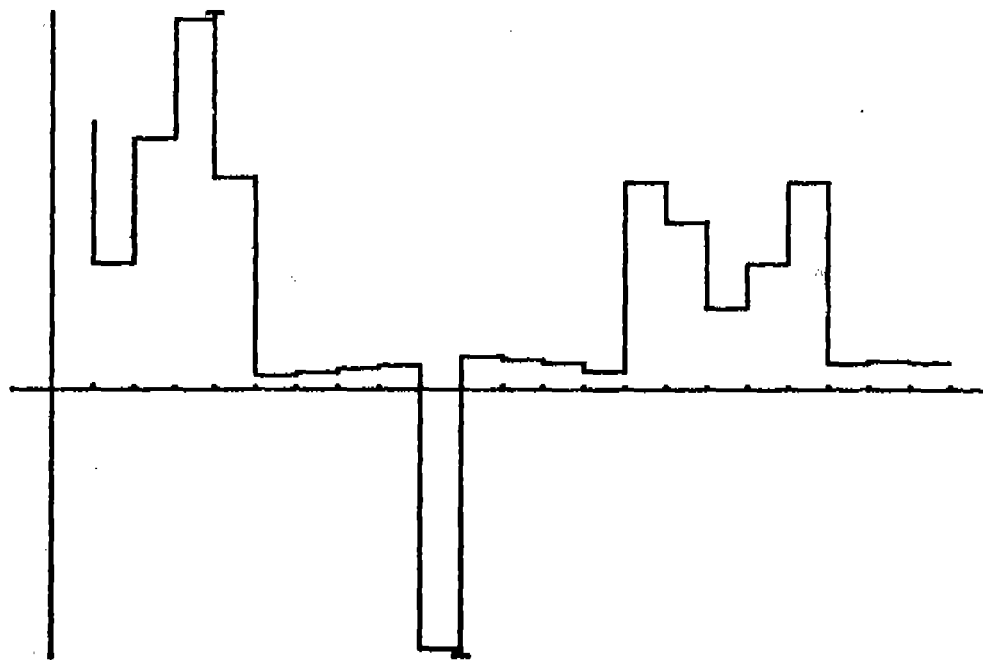
1,22

(* A CONTINUACION SE LISTAN LOS

DATOS ORIGINALES DE DISCO *)

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

**



GR4

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO DIBUJA LINEAS QUE
PARTIENDO DE LOS PUNTOS P1'(X1,0),P2'(X2,0),...
TERMINAN EN LOS PUNTOS P1(X1,Y1),P2(X2,Y2),...
RESPECTIVAMENTE . *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

w

##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

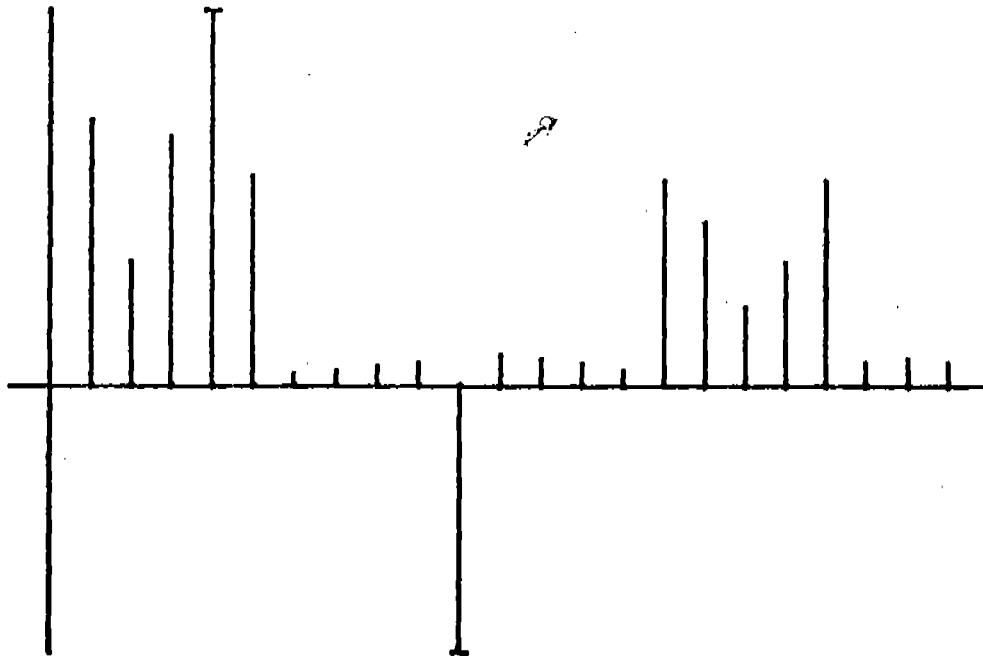
1,22

{* A CONTINUACION SE LISTAN LOS

DATOS ORIGINALES DE DISCO *}

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

##



GR5

* ALGORITMO DE GRAFICACION *

* OBSERVACION :

EL ALGORITMO UBICA LOS PUNTOS

P1(X1,Y1),P2(X2,Y2),...,PN(XN,YN)

EN EL PLANO FORMADO POR LAS COORDENADAS X E Y. *

* EJECUTA PRIMERO EL COMANDO LEESERIE *

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{*DESEA CONOCER EL TAMANO DE LA SERIE*}

NO

{* DESEA UN LISTADO?CONTESTE SI O NO *}

NO

w

##

LEESERIE

{* QUE ARCHIVO QUIERE LEER *}

DATOS1

{* DESEA CONOCER EL TAMANO DE LA SERIE *}

NO

{* DESEA UN LISTADO? CONTESTE SI O NO *}

SI

{* ESPECIFIQUE EL INTERVALO *}

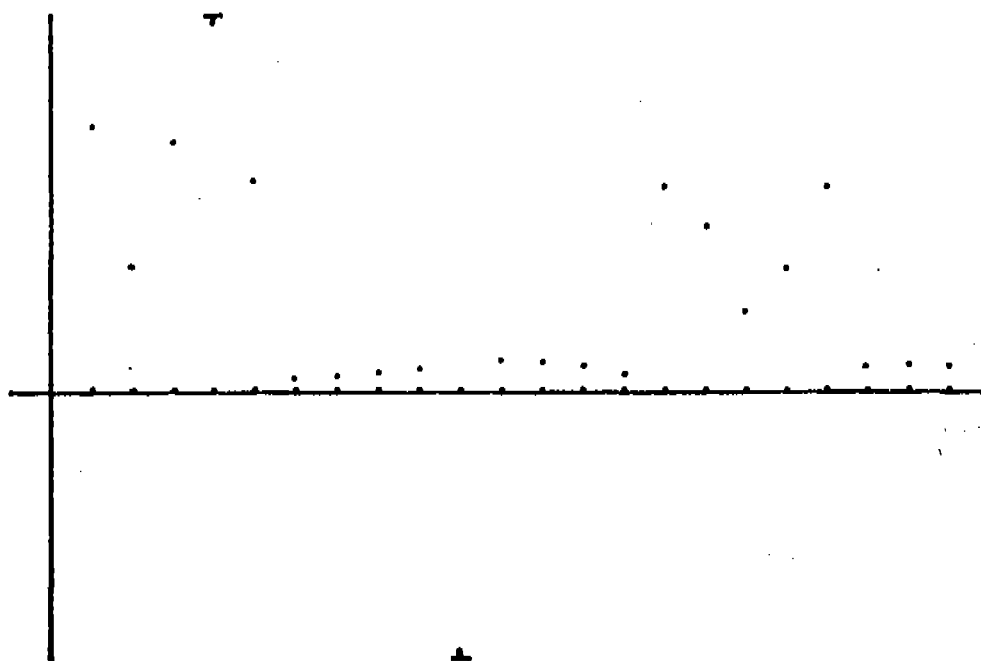
1,22

{* A CONTINUACION SE LISTAN LOS

DATOS ORIGINALES DE DISCO *}

DATO [1]	72.35000
DATO [2]	34.00000
DATO [3]	67.98000
DATO [4]	100.00000
DATO [5]	57.25000
DATO [6]	4.00000
DATO [7]	5.00000
DATO [8]	6.00000
DATO [9]	7.00000
DATO [10]	-70.00000
DATO [11]	9.00000
DATO [12]	8.00000
DATO [13]	7.00000
DATO [14]	5.00000
DATO [15]	56.00000
DATO [16]	45.00000
DATO [17]	22.00000
DATO [18]	34.00000
DATO [19]	56.00000
DATO [20]	7.00000
DATO [21]	8.00000
DATO [22]	7.00000

##



En los ejemplos se han usado diversos archivos ya sea en referencia a series, o a tablas. A continuación se listan los directorios respectivos.

7

1

10

FIN

#ET=6:59.9 PT=2.8 IO=8.0

L MDHF/DIRECTORIO/1

#FILE (ISAL)MDHF/DIRECTORIO/1 ON IIMAS

100	NOMBRE	TAM	DFIN
200	DATOS3	23	27
300	DATOS4	21	51
400	DATOS1	22	76
500	SERIE1	23	101
600	DISPON	0	0

#

L MDHF/DIRTAB/1

#FILE (ISAL)MDHF/DIRTAB/1 ON IIMAS

100	NOMBRE	TAM	DFIN	COLUM	FILAS	ORDEN
200	TABLA1	11	15	1	1	10
300	TABLA2	11	29	1	1	10
400	TABLA4	13	45	1	1	12
500	TABLA5	11	59	1	1	10
600	TABLA7	20	82	20	1	0
700	TABLA8	102	187	17	2	2
800	TABLA9	22	212	11	1	1
900	TABLA6	40	255	20	1	1
1000	TABL1	6	264	1	1	5
1100	TABL2	6	273	1	1	5
1200	TABL3	12	288	6	1	1

#

BYE
#END SESSION 1450 ET=32:39.8 PT=6.2 IO=10.8
#USER = ISAL 11:43:08 04/07/80

V. CONCLUSIONES .

Como se mencionó en la Introducción, el sistema SIMASET es un sistema de manejo de actividades de almacenamiento y de procesamiento numérico de la información.

Gracias a su estructura modular, es posible agregar o suprimir procedimientos de procesamiento numérico, algunos de los cuales no han sido desarrollados en lo que se refiere a los procesos de transformación de la información, aunque se ha previsto su implantación. Es por esto que existen sus archivos físicos en disco.

Los diversos algoritmos están agrupados en familias, y se encuentran almacenados en el directorio del IIMAS.

En cuanto a la utilización de SIMASET se prevee su empleo por parte de los investigadores en áreas como son el control de procesos, análisis estadístico, etc. Además se puede recomendar el uso de SIMASET como auxiliar en los cursos de Series de Tiempo que se imparten en la Maestría de Estadística e Investigación de Operaciones del IIMAS.

B I B L I O G R A F I A.

- [1] Anderson, T.W.; Time Series Analysis.
Wiley, 1970.

- [2] B 7000/6000 Binder Reference Manual
Forma # 5001456
Burroughs Corporation, 1970.

- [3] Gregor, Donald, J.; Alsol on the B6700
A complete Primer.
Volumen II, pag 151. Gregor Publishing Co., 1977.

- [4] Astrom, K.J.; Introduction to Stochastic Control.
Capitulo 6, párrafo 3. Academic Press, 1970.

AGRADECIMIENTO.

El autor del presente trabajo agradece la guía que le prestó el Dr. J.L. Farah, así como la ayuda ofrecida por el Dr. G. Sidhu en referencia a la organización del documento.

7

3

9

APENDICE A.

LISTADOS DE LOS PROGRAMAS

Los listados de los programas tienen un encabezado que se refiere a las secciones de los comandos que utilizan a los procedimientos de procesamiento.

```

100 BEGIN
200 COMMENT
300 -----
400
500 SS I M M A SS S EEE TTT
600 S S I I M M A A S S E E Y
700 S S I I M M A A A A S S E E T
800 S S I I M M A A S S E E T
900 S S I I M M A A S S E E T
1000 SS I M M A A SS EEE T

```

SISTEMA DE MANEJO DE SERIES DE TIEMPO

ESTE PROGRAMA NOS PERMITE REALIZAR DIVERSAS OPERACIONES QUE SE REFIEREN AL ALMACENAMIENTO Y RECUPERACION DE INFORMACION, A TRAVES DEL MANEJO DE ARCHIVOS FISICOS DE DISCO.

LAS OPERACIONES QUE SE REFIEREN AL MANEJO DE LA BASE DE DATOS SON:

- 1.- CREASERIE
- 2.- CREATABLA
- 3.- ESCRSERIE
- 4.- ESCR TABLA
- 5.- LFSERIE
- 6.- LFETABLA
- 7.- LYSISERIE
- 8.- LISTTABLA
- 9.- ACTUSERIE
- 10.- ACTUTABLA
- 11.- QYTASERIE
- 12.- QYTATABLA
- 13.- TERMINA

SE TRATA DE UN PROGRAMA INTERACTIVO.
PARA MAYOR INFORMACION VER EL CAPITULO III, SECCION 2.

```

FILE ARCH(KIND=DISK,MAXRECSIZE=1,BLOCKSIZE=90,
SAVEFACTOR=30),
FILE50(KTNC=REMOTE,MAXRECSIZE=600,UNITS=1),
DIRECTORIO(KIND=DISK,MAXRECSIZE=3,BLOCKSIZE=30,
SAVEFACTOR=30),
DIRUSER(KTND=DISK,FILETYPE=7),
TABLA(KIND=DISK,MAXRECSIZE=1,BLOCKSIZE=30,
SAVEFACTOR=30),
DIRTAB(KIND=DISK,MAXRECSIZE=6,BLOCKSIZE=30,
SAVEFACTOR=30),
VALORES(KIND=DISK,FILETYPE=7),
ENT(KIND=REMOTE),
SAL(KIND=REMOTE,MAXRECSIZE=22);

```

```

REAL C,D,I,J,K,K1,K2,L,N,NNN,M,P,F0C0,
YAM,DIFEC,II,JJ,TOTAL,Q,TOT,F,X1,X2,XX,YY,
NUM1,NUM2,NUM3;
ARRAY A,G,H,U(0:0),
IMAGENE(0:13),
NOMBRE1(0:39,0:2),
NOMBRE1(0:39,0:5),
ACEPTA(0:3),
AAA,CCCC(1:1);

```

```

00000100
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800
00005900
00006000
00006100
00006200
00006300

```

6400 EBCDIC ARRAY TIT,TITJLQ(0:14),
6500 DTIT(0:19),
6600 UCIO(3),
6700 COMANDO(0:13),
6800 VALOR(0:23),
6900 DTIT1,DTIT2(0:13);
7000

X

EBCDIC VALUE ARRAY

MSGC *** CUAL ES SU IDENTIFICADOR O *** ,48"00",Z1
" *** PASSWORD *** ,48"00",Z2
" *** CUAL ES SU CLAVE DE USUARIO *** ,48"00",Z3
" ** CUAL ES EL ORDEN DE LA TABLA *! ,48"00",Z4
" ** CON CUANTOS DATOS VA A TRABAJAR *! ,48"00",Z5
" ** A CONTINUACION SE LISTAN LOS ,48"00",Z6
" DATOS ORIGINALES DE DISCO *! ,48"00",Z7
" ** ESCRIBA SUS DATOS *! ,48"00",Z8
" ** EN CASO DE HABER ESCRITO EN DISCO ,48"00",Z9
" ** DESEA UN LISTADO? CONTESTE SI O NO *! ,48"00",Z10
" ** DE UN MAXIMO DE 6 CARACTERES PARA ,48"00",Z11
" EL NOMBRE DEL ARCHIVO A CREAR *! ,48"00",Z12
" ** YA EXISTE ESE ARCHIVO *! ,48"00",Z13
" ** DE UN BUENO NOMBRE DE ARCHIVO *! ,48"00",Z14
" ** QUE ARCHIVO QUIERE LEER *! ,48"00",Z15
" ** NO EXISTE ESE ARCHIVO EN DISCO *! ,48"00",Z16
" ** DESEA JD. ESCRIBIR EN DISCO *! ,48"00",Z17
" *** CONTESTE SI O NO *** ,48"00",Z18
" ** DESEA UC. LEER DE DISCO SU SERIE *! ,48"00",Z19
" *** CONTESTE SI O NO *** ,48"00",Z20
" ** DEFINA EL NUMERO DE COLUMNAS *! ,48"00",Z21
" ** DEFINA EL NUMERO DE FILAS *! ,48"00",Z22
" ** DESEA CREAR ARCHIVO DE SERIES *! ,48"00",Z23
" ** DESEA CREAR ARCHIVO DE TABLAS *! ,48"00",Z24
" ** YA CREO SU ARCHIVO DE SERIES *! ,48"00",Z25
" ** YA CREO SU ARCHIVO DE TABLAS *! ,48"00",Z26
" ** DEBIO HACER ANTES OPERACION DE I/O *! ,48"00",Z27
" ** DEBIO HACER OPERACION DE I/O *! ,48"00",Z28
" ** CUAL ARCHIVO VA A ACTUALIZAR *! ,48"00",Z29
" ** ESCRIBA: POSICION, DATO, NUEVO DATO *! ,48"00",Z30
" ** CUAL ARCHIVO DESEA SUPRIMIR *! ,48"00",Z31
" ** ESPECIFIQUE EL INTERVALO *! ,48"00",Z32
" ** ERROR EN LOS LIMITES DEL INTERVALO *! ,48"00",Z33
" ** SE ELIMINA DE LA LISTA DE USUARIOS *! ,48"00",Z34
" ** SE ELIMINAN SUS ARCHIVOS DE SERIES *! ,48"00",Z35
" ** SE ELIMINAN SUS ARCHIVOS DE TABLAS *! ,48"00",Z36
" ** DESEA CONOCER EL TAMAÑO DE LA SERIE *! ,48"00",Z37
" ** DEBIO CREAR ARCHIVO PREVIAMENTE *! ,48"00",Z38
" ** NO SE LE PUEDE ADMITIR AL SISTEMA *! ,48"00",Z39
" ** ERROR EN SU CLAVE DE USUARIO *! ,48"00",Z40
" ** ESPECIFIQUE EL INTERVALO EN N *! ,48"00",Z41
" ** DESEA CONOCER EL ORDEN DE LA TABLA *! ,48"00",Z42
" ** ERROR EN EL ORDEN N DE LA TABLA *! ,48"00",Z43
" ** ERROR EN EL DATO ALMACENADO *! ,48"00",Z44
"XXXXXX");

X

FORMAT F0(A6,T6,I6),
F1(A6,S(I6)),
F2(*NOMBRE TAM DFIN*),
F3(*NOMBRE TAM DFIN COLUM FILAS ORDEN*),
F4(X10,"DATO [",I4,"]",F2(5)),
F6(A6,3(I6));

X

POINTER AP1,AP2;

X

DEFINE FORI(C,D)=FOR I:=C STEP 1 UNTIL D DO N,
FORJ(C,D)=FOR J:=C STEP 1 UNTIL D DO N,
CONDICION(C)=CONDICION(C);

C0006400
C0006500
C0006600
C0006700
C0006800
C0006900
C0007000
C0007100
C0007200
C0007300
C0007400
C0007500
C0007600
C0007700
C0007800
C0007900
C0008000
C0008100
C0008200
C0008300
C0008400
C0008500
C0008600
C0008700
C0008800
C0008900
C0009000
C0009100
C0009200
C0009300
C0009400
C0009500
C0009600
C0009700
C0009800
C0009900
C0010000
C0010100
C0010200
C0010300
C0010400
C0010500
C0010600
C0010700
C0010800
C0010900
C0011000
C0011100
C0011200
C0011300
C0011400
C0011500
C0011600
C0011700
C0011800
C0011900
C0012000
C0012100
C0012200
C0012300
C0012400
C0012500
C0012600
C0012700
C0012800
C0012900

13100	READ(DIRUSER(I),F6,POINTER(ACEPTA[*]),	00013100
13200	FORJ(1,3) ACEPTA[J]);	00013200
13300	IF POINTER(ACEPTA[0])=UC FOR 4	00013300
13400	THEN	00013400
13500	IF ACEPTA[0]=0	00013500
13600	THEN BEGIN	00013500
13700	MENSAJE(33,1);	00013700
13800	FOFO:=2;	00013800
13900	WRITE(SAL,<"RR"/>);	00013900
14000	GO L2;	00014000
14100	END;	00014100
14200	END N;	00014200
14300		00014300
14400		00014400
14500		00014500
14600		00014600
14700		00014700
14800		00014800
14900		00014900
15000		00015000
15100		00015100
15200		00015200
15300		00015300
15400		00015400
15500		00015500
15600		00015600
15700		00015700
15800		00015800
15900		00015900
16000		00016000
16100		00016100
16200		00016200
16300		00016300
16400		00016400
16500		00016500
16600		00016600
16700		00016700
16800		00016800
16900		00016900
17000		00017000
17100		00017100
17200		00017200
17300		00017300
17400		00017400
17500		00017500
17600		00017600
17700		00017700
17800		00017800
17900		00017900
18000		00018000
18100		00018100
18200		00018200
18300		00018300
18400		00018400
18500		00018500
18600		00018500
18700		00018700
18800		00018800
18900		00018900
19000		00019000
19100		00019100
19200		00019200
19300		00019300
19400		00019400
19500		00019500

```

PROCEDURE INICIALIZA;
BEGIN
  REPLACE DTIT BY UC FOR 4,"/DIRECTORIO/",R
    FOR * DIGITS,".";
  REPLACE DIRECTORIO.TITLE BY DTIT;
  REPLACE TIT BY JC FOR 4,"/ARCHI/",R
    FOR * DIGITS,".";
  REPLACE ARCH1.TITLE BY TIT;
  REPLACE DTIT1 BY UC FOR 4,"/TABLA/",R FOR * DIGITS,".";
  REPLACE TABLA.TITLE BY DTIT1;
  REPLACE DTIT2 BY UC FOR 4,"/DIRTAB/",R FOR * DIGITS,".";
  REPLACE DIRTAB.TITLE BY DTIT2;
  REPLACE POINTER(U) BY UC FOR 4;
END;
PROCEDURE MENSAJE(XX,YY);
VALUE XX,YY;
REAL XX,YY;
FOR I:=0 STEP 1 UNTIL YY-1 DO
  WRITE(SAL,<A46>MSG((XX-1)+I)*48));
END;
PROCEDURE PREGUNTA;
BEGIN
  MENSAJE(4,1);
  READ(ENT,/,N);
  MENSAJE(21,1);
  READ(ENT,/,P);
  MENSAJE(22,1);
  READ(ENT,/,M);
END;
SECCION 2.3.3.1 COMANDO CREASERIE ABBREVIACION CES.
PROCEDURE CREASERIE;
BEGIN
  FOR I:=1 STEP 1 UNTIL 10 DO
  BEGIN
    READ(DIRUSER(I),F6,POINTER(ACEPTA[*]),FORJ(1,3) ACEPTA[J]);
    IF POINTER(ACEPTA[0])=UC FOR 4 AND ACEPTA[2]=0
    THEN
    BEGIN
      WRITE(DIRECTORIO(0),F2);
      LOCK(DIRECTORIO);
      WRITE(ARCHI(0),/,0.0);
      WRITE(ARCHI(1),/,1.0);
      LOCK(ARCHI);
      WRITE(SAL,<"* LOS ARCHIVOS FISICOS PARA SERIES SON :"/,
        X7,A15,/,X7,A20>,TIT,DTIT);
      ACEPTA[2]=1;
      WRITE(DIRUSER(I),F6,POINTER(ACEPTA[*]),FORJ(1,3) ACEPTA[J]);
    END
  END;

```


19600	ELSE	00019600
19700	IF POINTER(ACEPTA[1])=UC FOR 4 AND ACEPTA[2]=1	00019700
19800	THEN BEGIN	00019800
19900	MENSAJE(25,1);	00019900
20000	I:=12;	00020000
20100	END;	00020100
20200	END;	00020200
20300	WRITE(SAL, <"NN", />);	00020300
20400	END CREATSERIE;	00020400
20500	Z	00020500
20600	Z	00020600
20700	Z	00020700
20800	SECCION 2.3.3.2 COMANDO CREATABLA ABREVIACION CAT.	00020800
20900	PROCEDURE CREATABLA;	00020900
21000	BEGIN	00021000
21100	FOR I:=1 STEP 1 UNTIL 10 DO	00021100
21200	BEGIN	00021200
21300	READ(DIRUSER[I], F6, POINTER(ACEPTA[*]), FORJ(1,3) ACEPTA[J]);	00021300
21400	IF POINTER(ACEPTA[1])=UC FOR 4 AND ACEPTA[3]=0	00021400
21500	THEN	00021500
21600	BEGIN	00021600
21700	WRITE(DIRTAB[0], F3);	00021700
21800	LOCK(DIRTAB);	00021800
21900	WRITE(TABLA[0], /, 0.0);	00021900
22000	WRITE(TABLA[1], /, 1.0);	00022000
22100	LOCK(TABLA);	00022100
22200	WRITE(SAL, <"* LOS ARCHIVOS FISICOS PARA TABLAS SON :", /,	00022200
22300	X7, A14, /, X7, A14>, DTIT1, DTIT2);	00022300
22400	ACEPTA[3]=1;	00022400
22500	WRITE(DIRUSER[I], F6, POINTER(ACEPTA[*]), FORJ(1,3) ACEPTA[J]);	00022500
22600	END	00022600
22700	ELSE	00022700
22800	IF POINTER(ACEPTA[3])=UC FOR 4 AND ACEPTA[3]=1	00022800
22900	THEN BEGIN	00022900
23000	MENSAJE(26,1);	00023000
23100	I:=11;	00023100
23200	END;	00023200
23300	END;	00023300
23400	WRITE(SAL, <"NN", />);	00023400
23500	END CREATABLA;	00023500
23600	Z	00023600
23700	Z	00023700
23800	Z	00023800
23900	SECCION 2.3.3.13 COMANDO TERMINA ABREVIACION TER.	00023900
24000	PROCEDURE TERMINA;	00024000
24100	BEGIN	00024100
24200	REAL AUX1, AUX2;	00024200
24300	ARRAY ACEPTA[0:3];	00024300
24400	FORI(1,10)	00024400
24500	BEGIN	00024500
24600	READ(DIRUSER[I], F6, POINTER(ACEPTA[*]), FORJ(1,3) ACEPTA[J]);	00024600
24700	IF POINTER(ACEPTA[0])=UC FOR 4	00024700
24800	THEN BEGIN	00024800
24900	REPLACE POINTER(ACEPTA[0]) BY "CAMB";	00024900
25000	AUX1:=ACEPTA[2];	00025000
25100	AUX2:=ACEPTA[3];	00025100
25200	ACEPTA[1]=0;	00025200
25300	ACEPTA[2]=0;	00025300
25400	ACEPTA[3]=0;	00025400
25500	WRITE(DIRUSER[I], F6, POINTER(ACEPTA[*]),	00025500
25600	FORJ(1,3) ACEPTA[J]);	00025600
25700	I:=10;	00025700
25800	END;	00025800
25900	END;	00025900
26000	MENSAJE(34,1);	00026000
26100	IF AUX1=1	00026100
	THEN BEGIN	
	MENSAJE(35,1);	

```

26200 AFCH1.OPEN:=TRUE;
26300 CLOSE(ARCH1,PURGE);
26400 DIRECTORIO.OPEN:=TRUE;
26500 CLOSE(DIRECTORIO,PURGE);
26600 ENT;
26700 IF AUX2=1
26800 THEN BEGIN
26900 MENSAJE(36,1);
27000 TABLA.OPEN:=TRUE;
27100 CLOSE(TABLA,PURGE);
27200 DIRTAB.OPEN:=TRUE;
27300 CLOSE(DIRTAB,PURGE);
27400 ENT;
27500 FOR I(1,9)
27600 BEGIN
27700 READ(DIRUSER(I),F6,POINTER(ACEPTA(*)),
27800 FOR J(1,3) ACEPTA(J));
27900 READ(DIRUSER(I+1),F6,POINTER(ACEPTA1(*)),
28000 FOR J(1,3) ACEPTA1(J));
28100 IF POINTER(ACEPTA(0))="CAMB"
28200 THEN
28300 IF POINTER(ACEPTA1(0)) NEQ "CAMB"
28400 THEN BEGIN
28500 WRITEC(DIRUSER(I),F6,POINTER(ACEPTA1(*)),
28600 FOR J(1,3) ACEPTA1(J));
28700 WRITEC(DIRUSER(I+1),F6,POINTER(ACEPTA1(*)),
28800 FOR J(1,3) ACEPTA1(J));
28900 END;
29000 END;
29100 WRITE(SAL,<"N",/ >);
29200 END TERMINA;
29300
29400 X SECCION 2.3.3.7 COMANDO LISTSERIE ABBREVIACION LIS.
29500 X
29600 PROCEDURE LISTADJ(TOTAL,TAM,C,D,X1);
29700 REAL TOTAL,TAM,C,D;
29800 FILE X1;
29900 BEGIN
30000 LABEL L2;
30100 SWITCH FORMAT FF:=(X10,"DATO [",I1,"]",X3,F20.5),
30200 (X10,"DATO [",I2,"]",X2,F20.5),
30300 (X10,"DATO [",I3,"]",X1,F20.5),
30400 (X10,"DATO [",I4,"]",F20.5);
30500 JJ:=C-1;
30600 FOR I:=1 STEP 1 UNTIL TAM DO
30700 READ(X1[TOTAL-TAM-1+I],AAAC(I));
30800 IF C < 1 OR J > TAM
30900 THEN BEGIN
31000 MENSAJE(33,1);
31100 WRITE(SAL,<"EL TAMAÑO DE LA SERIE ES =",I6,>,TAM);
31200 GO L2;
31300 END;
31400 MENSAJE(6,2);
31500 FOR I:=C STEP 1 UNTIL D DO
31600 BEGIN
31700 JJ:=++1;
31800 IF JJ LEQ 9
31900 THEN WRITEC(SAL,FF(0),JJ,AAAC(I));
32000 IF JJ > 9 AND JJ LEQ 99
32100 THEN WRITEC(SAL,FF(1),JJ,AAAC(I));
32200 IF JJ > 99 AND JJ LEQ 999
32300 THEN WRITEC(SAL,FF(2),JJ,AAAC(I));
32400 IF JJ > 999 AND JJ LEQ 9999
32500 THEN WRITEC(SAL,FF(3),JJ,AAAC(I));
32600 END;
32700 L2:

```

```

00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027800
00027900
00028000
00028100
00028200
00028300
00028400
00028500
00028600
00028700
00028800
00028900
00029000
00029100
00029200
00029300
00029400
00029500
00029600
00029700
00029800
00029900
00030000
00030100
00030200
00030300
00030400
00030500
00030600
00030700
00030800
00030900
00031000
00031100
00031200
00031300
00031400
00031500
00031600
00031700
00031800
00031900
00032000
00032100
00032200
00032300
00032400
00032500
00032600
00032700

```

```

32800 END LISTADO;
32900
33000 Z SECCION 2.3.3.2 COMANDO LISTTABLA ABBREVIACION LIT.
33100 Z
33200 PROCEDURE LISTADOC(TOTAL,TAM,N,P,M,C,D);
33300 REAL N,P,M,TOTAL,TAM,C,D;
33400 BEGIN
33500 LABEL L2;
33600 ARRAY ARRTABE(1:M,1:2*P,C:N);
33700 CODIC ARRAY PENG(1:4);
33800 REAL CO,MM;
33900 REAL X,AUX1;
34000 MENSAJE(6,2);
34100 X:=TOTAL-TAM-1+(C*P*M);
34200 IF C > N
34300 THEN BEGIN
34400 MENSAJE(4,1);
34500 WRITE(SAL,<" ** EL ORDEN DE LA TABLA N =",I4,X1," *!">,N);
34600 GO L2;
34700 END;
34800 FOR K:=C STEP 1 UNTIL D DO
34900 BEGIN
35000 WRITE(SAL,<"/>,"LOS COEFICIENTES DE N =",I4,X2,"SON"/>,K);
35100 FOR I:=1 STEP 1 UNTIL M DO
35200 BEGIN
35300 REPLACE REVG BY "R",I FOR * DIGITS,".";
35400 WRITE(SAL,<A5>,RENG);
35500 AUX1:=0;
35600 FOR J:=2 STEP 2 UNTIL 2*P DO
35700 BEGIN
35800 ARRTABE(I,J-1,K):=AUX1:=**+1;
35900 READ(TABLA[X:=**+1],*,ARRTABE(I,J,K));
36000 END;
36100 MM:=0;
36200 FOR OO:=1 STEP 6 UNTIL 2*P DO
36300 BEGIN
36400 MM:=**+6;
36500 IF MM > 2*P THEN MM:=2*P;
36600 WRITE(SAL,<3(I4,X2,F16.4,X2)>, FORJ(OO,MM)
36700 ARRTABE(I,J,K));
36800 END;
36900 END;
37000 END;
37100 L2:
37200 END LISTADO;
37300
37400 Z PARTE DE LA SECCION 2.3.3.3 COMANDO ESCR SERIE ABBREVIACION ESE.
37500 Z
37600 PROCEDURE ESCRIBE1;
37700 BEGIN
37800 NOMBRE(1,0):=G(0);
37900 WRITE(ARCH1(TOTAL:=**+1),1,G(0));
38000 IF FOCO=0
38100 THEN BEGIN
38200 TAM:=0;
38300 WRITE(ARCH1(TOTAL:=**+1),<I6>,TAM);
38400 MENSAJE(8,1);
38500 WHILE NOT READ(ENT,/,A(0)) DO
38600 BEGIN
38700 WRITE(ARCH1(TOTAL:=**+1),*,A(0));
38800 TAM:=**+1;
38900 END;
39000 WRITE(ARCH1(TOTAL-TAM),<I6>,TAM);
39100 REWIND(ENT);
39200 END;
39300 IF FOCO=1

```

```

00032800
00032900
00033000
00033100
00033200
00033300
00033400
00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500
00035600
00035700
00035800
00035900
00036000
00036100
00036200
00036300
00036400
00036500
00036600
00036700
00036800
00036900
00037000
00037100
00037200
00037300
00037400
00037500
00037600
00037700
00037800
00037900
00038000
00038100
00038200
00038300
00038400
00038500
00038600
00038700
00038800
00038900
00039000
00039100
00039200
00039300

```

```

39400 THEN BEGIN
39500 FOR I:= 1 STEP 1 UNTIL TAM DO
39600 WRITE(ARCHI(TOTAL:=**+1),*,CCC(I));
39700 WRITE(ARCHI(TOTAL-TAM),<I6>,TAM);
39800 END;
39900 NOMBRE(II,1):=TAM;
40000 RESIZE(AAA,TAM);
40100 WRITE(ARCHI(1),/,TOTAL);
40200 DIREC:=TOTAL+1;
40300 NOMBRE(II,2):=DIREC;
40400 WRITE(DIRECTOR(II,1),FO, POINTER(NOMBRE(II,*)),
40500 NOMBRE(II,1),NOMBRE(II,2));
40600 II:=**+1;
40700 WRITE(ARCHI(DIREC),<I6>,II);
40800 LOCK(ARCHI);
40900 LOCK(DIRECTOR(1));
41000 MENSAJE(37,1);
41100 READ(ENT,<A2>,H(0));
41200 IF H(0)="SI"
41300 THEN
41400 WRITE(SAL,<" ** EL TAMANO DE LA SERIE ES=",I6,
41500 " **",/,>,TAM);
41600 REPLACE POINTER(H(0)) BY " " FOR 6;
41700 MENSAJE(9,2);
41800 READ(ENT,<A2>,H(0));
41900 IF H(0)="SI"
42000 THEN BEGIN
42100 MENSAJE(32,1);
42200 READ(ENT,/,C,D);
42300 LISTADO(TOTAL+1,TAM,C,D,ARCHI);
42400 END;
42500 END ESCRIBE1;
Z
42600
42700 PARTE DE LA SECCION 2.3.3.4 COMANDO ESCRITABLA ABREVIACION ETA.
Z
42800
42900 PROCEDURE ESCRIBE2(N,P,M);
43000 REAL N,P,M;
43100 BEGIN
43200 ARRAY ARRTAB(1:M,1:P,0:N);
43300 NOMBRE(II,0):=G(0);
43400 WRITE(TABLA(TOTAL:=**+1),1,G(0));
43500 TAM:=(N+1)*P*M;
43600 WRITE(TABLA(TOTAL:=**+1),<I6>,TAM);
43700 NOMBRE(II,1):=TAM;
43800 NOMBRE(II,5):=N;
43900 NOMBRE(II,3):=P;
44000 NOMBRE(II,4):=M;
44100 IF FOCO=0
44200 THEN BEGIN
44300 MENSAJE(8,1);
44400 FOR K:=0 STEP 1 UNTIL N DO
44500 FOR I:=1 STEP 1 UNTIL M DO
44600 FOR J:=1 STEP 1 UNTIL P DO
44700 BEGIN
44800 READ(ENT,/,ARRTAB(I,J,K));
44900 WRITE(TABLA(TOTAL:=**+1),*,ARRTAB(I,J,K));
45000 END;
45100 IF FOCO=1
45200 THEN BEGIN
45300 FOR I:=1 STEP 1 UNTIL TAM DO
45400 WRITE(TABLA(TOTAL:=**+1),*,CCC(I));
45500 END;
45600 WRITE(TABLA(1),/,TOTAL);
45700 DIREC:=TOTAL+1;
45800 NOMBRE(II,2):=DIREC;

```

```

00039400
00039500
00039600
00039700
00039800
00039900
00040000
00040100
00040200
00040300
00040400
00040500
00040600
00040700
00040800
00040900
00041000
00041100
00041200
00041300
00041400
00041500
00041600
00041700
00041800
00041900
00042000
00042100
00042200
00042300
00042400
00042500
00042600
00042700
00042800
00042900
00043000
00043100
00043200
00043300
00043400
00043500
00043600
00043700
00043800
00043900
00044000
00044100
00044200
00044300
00044400
00044500
00044600
00044700
00044800
00044900
00045000
00045100
00045200
00045300
00045400
00045500
00045600
00045700
00045800
00045900

```

46200	WRITE(DIRECTORY, F1, PCINTER(NOMBRE1(II,*)),	00046000
46100	FOPI(1,5) NOMBRE1(II,1));	00046100
46200	II:=I+1;	00046200
46300	WRITE(TABLA(DIRECT), <I6>, II);	00046300
46400	LOCK(TABLA);	00046400
46500	LOCK(DIRECT);	00046500
46600	MENSAJE(9,2);	00046600
46700	READ(FNT, <A2>, H(0));	00046700
46800	IF H(0) = "SI"	00046800
46900	THEN BEGIN	00046900
47000	MENSAJE(41,1);	00047000
47100	READ(ENT, /, C, 0);	00047100
47200	LISTADO1(TOTAL+1, TAM, N, P, M, C, D);	00047200
47300	END;	00047300
47400	END ESCRIBE2;	00047400
47500		00047500
47600	Z SECCION 2.3.3.3 COMANDO ESCRIBIR ABREVIACION ESE.	00047600
47700	Z SECCION 2.3.3.4 COMANDO ESCRIBIR TABLA ABREVIACION ETA.	00047700
47800	Z	00047800
47900	PROCEDURE ESCRIBIR(Q, X1, X2);	00047900
48000	REAL Q;	00048000
48100	FILE X1, X2;	00048100
48200	BEGIN	00048200
48300	LABEL L1, L2;	00048300
48400	IF Q=0	00048400
48500	THEN	00048500
48600	CONDICION(2);	00048600
48700	IF Q=1	00048700
48800	THEN	00048800
48900	CONDICION(3);	00048900
49000	READ(X1(1), /, TOTAL);	00049000
49100	READ(X1(TOTAL := I+1), /, FI);	00049100
49200	MENSAJE(11,2);	00049200
49300	L1:	00049300
49400	READ(FNT, 1, G(0));	00049400
49500	FOR I(1, II)	00049500
49600	BEGIN	00049600
49700	READ(X2(I-1), <A6>, NOMBRE1(I-1,0));	00049700
49800	IF NOMBRE1(I-1,0) = G(0)	00049800
49900	THEN BEGIN	00049900
50000	MENSAJE(13,2);	00050000
50100	I:=I+1;	00050100
50200	GO L1;	00050200
50300	END	00050300
50400	ELSE	00050400
50500	IF I=II	00050500
50600	THEN BEGIN	00050600
50700	CASE Q OF	00050700
50800	BEGIN	00050800
50900	BEGIN	00050900
51000	ESCRIBE1;	00051000
51100	I:=I+1;	00051100
51200	END;	00051200
51300	BEGIN	00051300
51400	PREGUNTA;	00051400
51500	ESCRIBE2(N, P, M);	00051500
51600	I:=I+1;	00051600
51700	END;	00051700
51800	END;	00051800
51900	END;	00051900
52000	END;	00052000
52100	L2:	00052100
52200	IF FOCO=0 THEN	00052200
52300	WRITE(SAL, <"NN", />);	00052300
52400	END ESCRIBIR;	00052400
52500		00052500

```

52500 Z SECCION 2.3.3.6 COMANDO LEETABLA ABREVIACION LAT. 00052600
52700 Z 00052700
52800 PROCEDURE LEE1(X1,X2); 00052800
52900 FILE X1,X2; 00052900
53000 BEGIN 00053000
53100 LABEL L2; 00053100
53200 CONDICION(3); 00053200
53300 MENSAJE(15,1); 00053300
53400 READ(FNT,1,GI(0)); 00053400
53500 READ(X1(0),/,K1); 00053500
53600 K1:=K1+1; 00053600
53700 READ(X1(K1),/,K2); 00053700
53800 IF K2-1=0 00053800
53900 THEN BEGIN 00053900
54000 MENSAJE(16,1); 00054000
54100 FOCO:=2; 00054100
54200 WRITE(SAL, <"NN",/ >); 00054200
54300 GO L2; 00054300
54400 END; 00054400
54500 FOR I:=1 STEP 1 UNTIL K2-1 DO 00054500
54600 BEGIN 00054600
54700 READ(X2(I),/,FORJ(0,5) NOMBRE1(I,J)); 00054700
54800 IF GTO1=NOMBRE1(I,0) 00054800
54900 THEN BEGIN 00054900
55000 T4:=NOMBRE1(I,1); 00055000
55100 T0TAL:=NOMBRE1(I,2); 00055100
55200 N:=NOMBRE1(I,3); 00055200
55300 P:=NOMBRE1(I,3); 00055300
55400 M:=NOMBRE1(I,4); 00055400
55500 RESTZE(AAA,TAM); 00055500
55600 FOR J:=1 STEP 1 UNTIL TAM DO 00055600
55700 READ(X1(TOTAL-TAM-I,J),/,AAAJ); 00055700
55800 I:=K2+1; 00055800
55900 END 00055900
56000 ELSE IF I=K2-1 00056000
56100 THEN BEGIN 00056100
56200 MENSAJE(16,1); 00056200
56300 FOCO:=2; 00056300
56400 I:=K2+1; 00056400
56500 WRITE(SAL, <"NN",/ >); 00056500
56600 GO L2; 00056600
56700 END; 00056700
56800 END; 00056800
56900 MENSAJE(42,1); 00056900
57000 READ(FNT, <A2>,HE(0)); 00057000
57100 IF HE(0)="SI" 00057100
57200 THEN 00057200
57300 WRITE(SAL, <" ** EL ORDEN N DE LA TABLA ES =",I4," *!">,N); 00057300
57400 MENSAJE(10,1); 00057400
57500 READ(FNT, <A2>,HE(0)); 00057500
57600 IF HE(0)="SI" 00057600
57700 THEN BEGIN 00057700
57800 MENSAJE(41,1); 00057800
57900 READ(FNT,/,C,D); 00057900
58000 LISTADD1(TOTAL,TAM,N,P,M,C,D); 00058000
58100 END; 00058100
58200 L2: 00058200
58300 IF FOCO=0 THEN 00058300
58400 WRITE(SAL, <"NN",/ >); 00058400
58500 END LEE1; 00058500
58600 00058600

```

```

58700 Z SECCION 2.3.3.5 COMANDO LEESERIE ABREVIACION LES. 00058700
58800 Z 00058800
58900 PROCEDURE LEE(X1,X2); 00058900
59000 FILE X1,X2; 00059000
59100 BEGIN 00059100

```

59200	LABEL L2;	00059200
59300	CONEXION(2);	00059300
59400	MENSAJE(15,1);	00059400
59500	READ(FNT,1,G(0));	00059500
59600	READ(X1(0),/,X1);	00059600
59700	K1:=*+1;	00059700
59800	READ(X1(K1),/,K2);	00059800
59900	IF K2-1=0	00059900
60000	THEN BEGIN	00060000
60100	MENSAJE(16,1);	00060100
60200	FOCO:=2;	00060200
60300	WRITE(SAL, <"RR"/>);	00060300
60400	GO L2;	00060400
60500	END;	00060500
60600	FOR I:=1 STEP 1 UNTIL K2-1 DO	00060600
60700	BEGIN	00060700
60800	READ(X2(I),FO) FOR J:=0 STEP 1 UNTIL 2 DO	00060800
60900	NOMBRE(I,J);	00060900
61000	IF G(0)=NOMBRE(I,0)	00061000
61100	THEN BEGIN	00061100
61200	TAM:=NOMBRE(I,1);	00061200
61300	TOTAL:=NOMBRE(I,2);	00061300
61400	RFSIZE(AAA,TAM);	00061400
61500	FOR J:=1 STEP 1 UNTIL TAM DO	00061500
61600	READ(X1(TOTAL-TAM-1+J),/,AAA(J));	00061600
61700	T:=K2+1;	00061700
61800	END	00061800
61900	ELSE IF I=K2-1 THEN	00061900
62000	BEGIN	00062000
62100	MENSAJE(16,1);	00062100
62200	FOCO:=2;	00062200
62300	I:=K2+1;	00062300
62400	WRITE(SAL, <"RR"/>);	00062400
62500	GO L2;	00062500
62600	END;	00062600
62700	END;	00062700
62800	MENSAJE(37,1);	00062800
62900	READ(FNT, <A2>, H(0));	00062900
63000	IF H(0)="SI"	00063000
63100	THEN	00063100
63200	WRITE(SAL, <" * EL TAMANO DE LA SERIE ES =", I6,	00063200
63300	X2, " *"/>, TAM);	00063300
63400	REPLACE POINTER(H(0)) BY " " FOR 6;	00063400
63500	MENSAJE(10,1);	00063500
63600	READ(FNT, <A2>, H(0));	00063600
63700	IF H(0)="SI"	00063700
63800	THEN BEGIN	00063800
63900	MENSAJE(32,1);	00063900
64000	READ(FNT,/,C,0);	00064000
64100	LISTADG(TOTAL,TAM,C,0,ARCH1);	00064100
64200	END;	00064200
64300	L2:	00064300
64400	IF FOCO=0 THEN	00064400
64500	WRITE(SAL, <"RR"/>);	00064500
64600	END LEE;	00064600
64700		00064700
64800	SECCION 2.3.3.9 COMANDO ACTUSERIE ABBREVIACION ACS.	00064800
64900	SECCION 2.3.3.10 COMANDO ACTUTABLA ABBREVIACION ACT.	00064900
65000		00065000
65100	PROCEDURE ACTUALIZA(QQ,X1,X2,X,XX);	00065100
65200	FILE X1,X2;	00065200
65300	REAL QQ,XX;	00065300
65400	ARRAY X(0,0);	00065400
65500	BEGIN	00065500
65600	LABEL L2;	00065600
65700	SWITCH FORMAT FF:=(A6,2(I6)),(A6,5(I6));	00065700

Z
Z
Z
Z

```

65900 IF Q0=0 THEN CONDICION(2);
65950 IF Q0=1 THEN CONDICION(3);
66000 MENSAJE(29,1);
66100 READ(FNT,1,GC(3));
66200 READ(X1(1),/,X1);
66300 K1:=*+1;
66400 READ(X1(K1),/,K2);
66500 IF K2-1=0
66600 THEN BEGIN
66700 MENSAJE(16,1);
66800 GO L2;
66900 END;
67000 FOR I:=1 STEP 1 UNTIL K2-1 DO
67100 BEGIN
67200 READ(X2(I),FFI(0),FORJ(0,XX) X(I,J));
67300 IF X(I,0)=3(0)
67400 THEN BEGIN
67500 IF XX=2
67600 THEN BEGIN
67700 TAM:=X(I,1);
67800 TOTAL:=X(I,2);
67900 END;
68000 IF XX=5
68100 THEN BEGIN
68200 TAM:=X(I,1);
68300 TOTAL:=X(I,2);
68400 N:=X(I,5);
68500 P:=X(I,3);
68600 M:=X(I,4);
68700 END;
68800 BEGIN
68900 LABEL L1;
69000 REAL AUX(1)ATO,NDATO,POSICION,LUGAR,N1,M1;
69100 L1:
69200 IF Q0=0
69300 THEN BEGIN
69400 MENSAJE(3,1);
69500 READ(ENT,/,POSICION,DATO,NDATO);
69600 END;
69700 IF Q0=1
69800 THEN BEGIN
69900 WRITE(SAL,<*** ESCRIBA SEPARADO POR COMAS :/,/
70000 X7,"MATRIZ DONDE ESTA EL DATO",/,X7,
70100 "RENGLON DE LA MATRIZ",/,X7,"COLUMNA DE LA",
70200 " MATRIZ",/,X7,"EL DATO ACTUAL ALMACENADO",/,
70300 X7,"Y EL NUEVO DATO *!*>);
70400 READ(ENT,/,N1,M1,POSICION,DATO,NDATO);
70500 POSICION:=*+(N1*M+M1-1)*P;
70600 END;
70700 LUGAR:=TOTAL-TAM+POSICION-1;
70800 READ(X1(LUGAR),/,AUX);
70900 IF DATO=AUX
71000 THEN BEGIN
71100 WRITE(X1(LUGAR),/,NDATO);
71200 LOCK(X1);
71300 END;
71400 ELSE
71500 IF DATO NEQ AUX
71600 THEN BEGIN
71700 MENSAJE(44,1);
71800 IF Q0=0
71900 THEN
72000 WRITE(SAL,<EL DATO EN LA POSICION",I6,/, "ES",F24.8>,
72100 POSICION,AUX);
72200 IF Q0=1
72300 THEN BEGIN

```

```

00055800
00055900
00056000
00056100
00056200
00056300
00056400
00056500
00056600
00056700
00056800
00056900
00057000
00057100
00057200
00057300
00057400
00057500
00057600
00057700
00057800
00057900
00058000
00058100
00058200
00058300
00058400
00058500
00058600
00058700
00058800
00058900
00059000
00059100
00059200
00059300
00059400
00059500
00059600
00059700
00059800
00059900
00070000
00070100
00070200
00070300
00070400
00070500
00070600
00070700
00070800
00070900
00071000
00071100
00071200
00071300
00071400
00071500
00071600
00071700
00071800
00071900
00072000
00072100
00072200
00072300

```



```

72400      POSICION:=*(N1*M+M1-1)*P;
72500      WRITE(SAL, <"EN LA MATRIZ", I6, /,
72600          "RENGLON", I6, /,
72700          "COLUMNA", I6, /,
72800          "EL VALOR DEL DATO ES", F24.8>,
72900          N1, M1, POSICION, AUX);
73000      END;
73100      GO L1;
73200      EV:
73300      END;
73400      I:=K2+1;
73500      END;
73600      ELSE IF I=<2-1
73700      THEN BEGIN
73800      MENSAJE(16, 1);
73900      I:=K2+1;
74000      GO L2;
74100      END;
74200      END;
74300      MENSAJE(10, 1);
74400      READCENT(<A2>, HI0));
74500      CASE QQ OF
74600      BEGIN
74700      IF HI01="SI"
74800      THEN BEGIN
74900      MENSAJE(32, 1);
75000      READCENT(7, C, D);
75100      RESIZE(CAAA, TAM);
75200      LTSTABDC(TOTAL, TAM, C, D, ARCH1);
75300      END;
75400      IF HI01="SI"
75500      THEN BEGIN
75600      MENSAJE(41, 1);
75700      READCENT(7, C, D);
75800      LTSTABDC(TOTAL, TAM, N, P, M, C, D);
75900      END;
76000      END;
76100      L2:
76200      IF FOCO=0 THEN
76300      WRITE(SAL, <"NN", />);
76400      END ACTUALIZA;
Z
Z SECCION 2.3.3.11 COMANDO QITASERIE ABREVIACION QIS.
Z SECCION 2.3.3.12 COMANDO QITATABLA ABREVIACION QIT.
Z
PROCEDURE QUITA(QQ, X1, X2, X, XX);
FILE X1, X2;
REAL QQ, XX;
ARRAY X(0, 0);
BEGIN
LABEL L2;
REAL JJ, LUGAR, CONT;
SWITCH FORMAT FF:=(A6, 2(I6)), (A6, 5(I6));
IF QQ=0
THEN
CONDICION(2);
IF QQ=1
THEN
CONDICION(3);
JJ:=0;
LUGAR:=0;
MENSAJE(31, 1);
READ(FNT, 1, C(1));
READ(X1(0), /, K1);
K1:=*+1;
READ(X1(K1), /, K2);

```

```

00072400
00072500
00072600
00072700
00072800
00072900
00073000
00073100
00073200
00073300
00073400
00073500
00073600
00073700
00073800
00073900
00074000
00074100
00074200
00074300
00074400
00074500
00074600
00074700
00074800
00074900
00075000
00075100
00075200
00075300
00075400
00075500
00075600
00075700
00075800
00075900
00076000
00076100
00076200
00076300
00076400
00076500
00076600
00076700
00076800
00076900
00077000
00077100
00077200
00077300
00077400
00077500
00077600
00077700
00077800
00077900
00078000
00078100
00078200
00078300
00078400
00078500
00078600
00078700
00078800
00078900

```

```

79000 IF K2-1=0
79100 THEN BEGIN
79200 MENSAJE(15,1);
79300 GO L2;
79400
79500 END;
79600 FOR I:=1 STEP 1 UNTIL K2-1 DO
79700 BEGIN
79800 READ(X2[I],FF[Q],FORJ(,XX) X[I,J]);
79900 IF X[I,0]=3[0]
80000 THEN BEGIN
80100 REPLACE POINTER(XE I,*) BY "DISPON";
80200 X[I,1]:=X[I,2]:=0;
80300 WRITE(X2[I],FF[Q],FORJ(,XX) X[I,J]);
80400 LOCK(X2);
80500 I:=K2+1;
80600 END
80700 ELSE
80800 IF I=K2-1
80900 THEN BEGIN
81000 MENSAJE(16,1);
81100 GO L2;
81200 END;
81300 END;
81400 CNT:=0;
81500 FOR I:=1 STEP 1 UNTIL K2-1 DO
81600 BEGIN
81700 READ(X2[I],FF[Q],FORJ(,XX) X[I,J]);
81800 IF X[I,0] NEG "DISPON"
81900 THEN BEGIN
82000 CNT:=++1;
82100 TAM:=X[I,1];
82200 LUGAR:=++3+TAM;
82300 END;
82400 IF X[I,0]="DISPON"
82500 THEN BEGIN
82600 FOR JJ:=I+1 STEP 1 UNTIL K2-1 DO
82700 BEGIN
82800 READ(X2[JJ],FF[Q],FORJ(,XX) X[JJ,J]);
82900 IF X[JJ,0] NEG "DISPON"
83000 THEN BEGIN
83100 TAM:=X[JJ,1];
83200 TOTAL:=X[JJ,2];
83300 BEGIN
83400 ARRAY AX(TOTAL-TAM:TAM-1);
83500 FOR K:=TOTAL-TAM STEP 1 UNTIL TOTAL-1 DO
83600 READ(X[K],*,AUX[K]);
83700 CONT:=+1;
83800 WRITE(X[LUGAR:=++1],<IS>,CONT);
83900 WRITE(X[LUGAR:=++1],1,X[JJ,0]);
84000 WRITE(X[LUGAR:=++1],<IS>,TAM);
84100 FOR K:=TOTAL-TAM STEP 1 UNTIL TOTAL-1 DO
84200 WRITE(X[LUGAR:=++1],*,AUX[K]);
84300 DIREC:=LUGAR+1;
84400 X[JJ,2]:=DIREC;
84500 WRITE(X2[JJ-1],FF[Q],FORJ(,XX) X[JJ,J]);
84600 IF JJ=K2-1
84700 THEN BEGIN
84800 REPLACE POINTER(XEJJ,*) BY "DISPON";
84900 X[JJ,1]:=X[JJ,2]:=0;
85000 WRITE(X2[JJ],FF[Q],FORJ(,XX) X[JJ,J]);
85100 END;
85200 LOCK(X2);
85300 END;
85400 END;
85500 END;
WRITE(X1[0],/,LUGAR);

```

```

00079000
00079100
00079200
00079300
00079400
00079500
00079600
00079700
00079800
00079900
00080000
00080100
00080200
00080300
00080400
00080500
00080600
00080700
00080800
00080900
00081000
00081100
00081200
00081300
00081400
00081500
00081600
00081700
00081800
00081900
00082000
00082100
00082200
00082300
00082400
00082500
00082600
00082700
00082800
00082900
00083000
00083100
00083200
00083300
00083400
00083500
00083600
00083700
00083800
00083900
00084000
00084100
00084200
00084300
00084400
00084500
00084600
00084700
00084800
00084900
00085000
00085100
00085200
00085300
00085400
00085500

```

85600	CONTI:=**+1;	00085600
85700	WRITE(X1(LUGAR:=**+1),<I6>,CONT);	00085700
85800	LOCK(X1);	00085800
85900	WRITE(SAL,<** SE ELIMINA EL ARCHIVO **>);	00085900
86000	WRITE(SAL,<X9,A6>,G[0]);	00086000
86100	I:=K2+1;	00086100
86200	END;	00086200
86300	END;	00086300
86400	L:	00086400
86500	IF F000=0 THEN	00086500
86600	WRITE(SAL,<"GR",/ >);	00086600
86700	END QUITA;	00086700
86800	Z	00086800
86900	Z	00086900
87000	\$DUMPINFO "ISAL/SIMASET/INFO";	00087000
87100	Z	00087100
87200	Z	00087200
87300	PROCEDURE EJECUTA;	00087300
87400	EXTERNAL;	00087400
87500	Z	00087500
87600	Z	00087600
87700	Z	00087700
87800	EJECUTA:	00087800
87900	END.	00087900

100	BEGIN		
200	FILE DIRUSER(KIND=DISK,MAXRECSIZE=4,BLOCKSIZE=60,		00000100
300	SAVEFACTOR=30),		00000200
400	SAL(KIND=REMOTE);		00000300
410	REAL I,J;		00000400
420	FORMAT F1("CLAVE NUMERO SERIE TABLA"),		00000410
430	F2(A6,3(I6));		00000420
500	ARRAY TITC(3);		00000430
600	REPLACE POINTER(TITC:1) BY "DISP";		00000440
700	WRITE(SAL,<"CENTRAL ARCHIVO">);		00000450
710	WRITE(DIRUSER(1),F1);		00000460
800	FOR I:=1 STEP 1 UNTIL 10 DO		00000470
900	WRITE(DIRUSER(I),F2,POINTER(TITC+1)),FOR J:= 1 STEP 1 UNTIL 3		00000480
910	30 TIT(J));		00000490
1000	LOCK(DIRUSER,*);		00000500
1100	END.		00000510
			00000520
			00000530
			00000540
			00000550
			00000560
			00000570
			00000580
			00000590
			00000600
			00000610
			00000620
			00000630
			00000640
			00000650
			00000660
			00000670
			00000680
			00000690
			00000700
			00000710
			00000720
			00000730
			00000740
			00000750
			00000760
			00000770
			00000780
			00000790
			00000800
			00000810
			00000820
			00000830
			00000840
			00000850
			00000860
			00000870
			00000880
			00000890
			00000900
			00000910
			00000920
			00000930
			00000940
			00000950
			00000960
			00000970
			00000980
			00000990
			00001000

100	CREASERIE	0	CES
200	CREATABLA	1	CAT
300	ESCRSERIE	2	ESC
400	ESCRTABLA	3	ETA
500	LEESERIE	4	LES
600	LEETABLA	5	LAT
700	LISTSERIE	6	LIS
800	LISTTABLA	7	LIT
900	ACTUSERIE	8	ACS
1000	ACTUTABLA	9	ACT
1100	QUITASERIE	10	QIS
1200	QUITATABLA	11	QIT
1300	TERMINA	12	TER
1400	SEN/1(MAX)	13	SN1
1500	SEN/2(MIN)	14	SN2
1600	SEN/3(MEDIA)	15	SN3
1700	SEN/4(NORM2)	16	SN4
1800	SEN/5(NORMINF)	17	SN5
1900	SEN/6(NORM1)	18	SN6
2000	SEN/7(DEF SVIA)	19	SN7
2100	SES/1(SUMA)	20	SE1
2200	SES/2(MULT)	21	SE2
2300	SES/3(CONVOL)	22	SE3
2400	TAT/1(SUMA)	23	TT1
2500	TAT/2(PREDICT)	24	TT2
2600	TAT/3(CONTRCL)	25	TT3
2700	TAN/1(MAX)	26	TN1
2800	TAN/2(MIN)	27	TN2
2900	TAN/3(MEDIA)	28	TN3
3000	SS/1(FILTRA)	29	S11
3100	SS/2(FILTRA)	30	S12
3200	SS/3(FILTRA)	31	S13
3300	SS/1(ESPECTRO)	32	S21
3400	SS/2(ESPECTRO)	33	S22
3500	SS/3(ESPECTRO)	34	S23
3600	SS/1(SUAVIZA)	35	S31
3700	SS/2(SUAVIZA)	36	S32
3800	SS/3(SUAVIZA)	37	S33
3900	SS/(ESCALAR)	38	SE
4000	SET/1(IDENT)	39	ST1
4100	SET/2(IDENT)	40	ST2
4200	SET/3(IDENT)	41	ST3
4300	SET/1(HISTO)	42	ST4
4400	SET/2(HISTO)	43	ST5
4500	SET/3(HISTO)	44	ST5
4600	TAS/1(SIMULA)	45	TS1
4700	TAS/2(SIMULA)	46	TS2
4800	TAS/3(SIMULA)	47	TS3
4900	GRA/1(COINT)	48	GR1
5000	GRA/2(HISTO)	49	GR2
5100	GRA/3(HISTO)	50	GR3
5200	GRA/4(LINEAS)	51	GR4
5300	GRA/5(PUNTOS)	52	GR5

```

100  & SET LEVEL 3
200  & SET AUTOBIND
300  & HOST IS OBJECT/SIMASET
400  I
500  &LOADINFO *ISAL/SIMASET/INFO*
600  PROCEDURE SENC(INICIALIZA,F,G);
610  PROCEDURE INICIALIZA;
620  PROCEDURE F(X1,X2);
630  FILE X1,X2;
640  FORMAL;
650  REAL G;
660  EXTERNAL;
670  Z
680  Z
690  PROCEDURE SES(INICIALIZA,F,G,S);
700  PROCEDURE INICIALIZA;
710  PROCEDURE F(X1,X2);
720  FILE X1,X2;
730  FORMAL;
740  PROCEDURE G(Q,X1,X2);
750  REAL S;
760  FILE X1,X2;
770  FORMAL;
780  REAL S;
790  EXTERNAL;
800  Z
810  Z
820  PROCEDURE TAN(INICIALIZA,F,G);
830  PROCEDURE INICIALIZA;
840  PROCEDURE F(X1,X2);
850  FILE X1,X2;
860  FORMAL;
870  REAL G;
880  EXTERNAL;
890  Z
900  Z
910  PROCEDURE TAT(INICIALIZA,F,G,S);
920  PROCEDURE INICIALIZA;
930  PROCEDURE F(X1,X2);
940  FILE X1,X2;
950  FORMAL;
960  PROCEDURE G(Q,X1,X2);
970  REAL S;
980  FILE X1,X2;
990  FORMAL;
1000 REAL S;
1010 EXTERNAL;
1020 Z
1030 Z
1040 PROCEDURE SS(INICIALIZA,F,G,S);
1050 PROCEDURE INICIALIZA;
1060 PROCEDURE F(X1,X2);
1070 FILE X1,X2;
1080 FORMAL;
1090 PROCEDURE G(Q,X1,X2);
1100 REAL S;
1110 FILE X1,X2;
1120 FORMAL;
1130 REAL S;
1140 EXTERNAL;
1150 Z
1160 Z
1170 PROCEDURE SET(INICIALIZA,F,G,S);

```

```

00000100
00000200
00000300
00000400
00000500
00000600
00000510
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170

```

```

1180  PROCEDURE INICIALIZA;
1190  PROCEDURE F(X1,X2);
1200  FILE X1,X2;
1210  FORMAL;
1220  PROCEDURE G(Q,X1,X2);
1230  REAL Q;
1240  FILE X1,X2;
1250  FORMAL;
1260  REAL S;
1270  EXTERNAL;
1280
1290  X
1300  PROCEDURE T(INICIALIZA,F,G,S);
1310  PROCEDURE INICIALIZA;
1320  PROCEDURE F(X1,X2);
1330  FILE X1,X2;
1340  FORMAL;
1350  PROCEDURE G(Q,X1,X2);
1360  REAL Q;
1370  FILE X1,X2;
1380  FORMAL;
1390  REAL S;
1400  EXTERNAL;
1410
1420  X
1430  PROCEDURE GR(INICIALIZA,F,Q);
1440  PROCEDURE INICIALIZA;
1450  PROCEDURE F(X1,X2);
1460  FILE X1,X2;
1470  FORMAL;
1480  REAL Q;
1490  EXTERNAL;
1500 ]
1600 PROCEDURE EJECUTA;
1700 BEGIN
1710 LABEL AFUERA;
1800 WRITE(SAL,<"*** CUAL ES SU CLAVE DE USUARIO ***">);
1810 READ(ENT,<A4>,UC);
1820 WRITE(SAL,<"*** CUAL ES SU IDENTIFICADOR O PASSWORD ***">);
1830 READ(ENT,<F>);
1840 INICIALIZA;
1850 FORI(1,11)
1860 BEGIN
1870 IF I=11
1880 THEN BEGIN
1890 MENSAJE(39,1);
1900 GO AFUERA;
1910 END;
1920 READ(DIRUSER(I),F6,POINTER(ACEPTA(I)),FORJ(1,3) ACEPTA(J));
1930 IF POINTER(ACEPTA(0))="DISP" OR POINTER(ACEPTA(0))="CMB"
1940 THEN BEGIN
1950 REPLACE POINTER(ACEPTA(0)) BY UC FOR 4;
1960 IF ACEPTA(1)=0 THEN ACEPTA(1)=R
1970 ELSE ACEPTA(1):=ACEPTA(1);
1980 ACEPTA(2):=0;
1990 ACEPTA(3):=0;
2000 WRITE(DIRUSER(I),F6,POINTER(ACEPTA(*)),FORJ(1,3) ACEPTA(J));
2010 I:=12;
2020 END
2030 ELSE
2040 IF POINTER(ACEPTA(0))=UC FOR 4
2050 THEN IF ACEPTA(1)=R
2060 THEN I:=12
2070 ELSE BEGIN
2080 MENSAJE(40,1);
2090 GO AFUERA;

```

```

00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001600
00001700
00001710
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090

```

```

2100 END;
2110 WRITE(SAL,</,"*** BIENVENIDO AL SISTEMA SIMASET ***">);
2120 WRITE(SAL,<X12,"***COMANDOS***",/,>);
2130 NNN:=52;
2140 BEGIN
2150 LABEL OTRO;
2160 ARRAY COMANDOSEC:NNN,0:3];
2165 FOR I:=0 STEP 1 UNTIL NNN DO
2166 READ(CVALORES,4,COMANDOS[I,*]);
2170 OTRO:
2180 FOCC:=0;
2190 READ(ENT,14,IMAGEN);
2200 AP1:=POINTER(IMAGEN);
2210 WRITE(SAL,<" ">);
2220 SCAN AP1:AP1 FOR NUM1:80 UNTIL NEG " ";
2230 SCAN AP2:AP1 FOR NUM2:NUM1 UNTIL EQL " ";
2240 NUM3:=NUM1-NUM2;
2250 REPLACE COMANDO BY AP1 FOR NUM3," " FOR (14-NUM3);
2260 IF COMANDO NEG " IN" THEN
2270 BEGIN
2280 FOR I:=0 STEP 1 UNTIL NNN DO
2290 BEGIN
2300 REPLACE VALOR BY " " FOR 24;
2310 REPLACE VALOR BY POINTER(COMANDOS[I,*]) FOR 24;
2320 IF VALOR[0]=COMANDOS[0] FOR 14 OR COMANDOS[0]=VALOR[21] FOR 3
2330 THEN BEGIN
2340 Q:=INTEGER(VALOR[15],3);
2350 CASE C OF
2360 BEGIN
2370 CREATSERIE;
2380 CREATABLA;
2390 ESCRIBE(C,ARCH1,DIRECTORIO);
2400 ESCRIBE(1,TABLA,DIRTAB);
2410 LEE(ARCH1,DIRECTORIO);
2420 LEE(TABLA,DIRTAB);
2430 BEGIN
2440 MENSAJE(27,1);
2450 WRITE(SAL,<" NN",/,>);
2460 END;
2470 BEGIN
2480 MENSAJE(28,1);
2490 WRITE(SAL,<" NN",/,>);
2500 END;
2510 ACTUALIZAC(C,ARCH1,DIRECTORIO,NOMBRE,2);
2520 ACTUALIZAC(1,TABLA,DIRTAB,NOMBRE1,5);
2530 QUITA(C,ARCH1,DIRECTORIO,NOMBRE,2);
2540 QUITA(1,TABLA,DIRTAB,NOMBRE1,5);
2550 TERMINA;
2560 SEND(INICIALIZA,LEE,0);
2570 SEND(INICIALIZA,LEE,1);
2580 SEND(INICIALIZA,LEE,2);
2590 SEND(INICIALIZA,LEE,3);
2600 SEND(INICIALIZA,LEE,4);
2610 SEND(INICIALIZA,LEE,5);
2620 SEND(INICIALIZA,LEE,6);
2630 SEND(INICIALIZA,LEE,ESCRIBE,0);
2640 SEND(INICIALIZA,LEE,ESCRIBE,1);
2650 SEND(INICIALIZA,LEE,ESCRIBE,2);
2660 TAT(INICIALIZA,LEE1,ESCRIBE,0);
2670 TAT(INICIALIZA,LEE1,ESCRIBE,1);
2680 TAT(INICIALIZA,LEE1,ESCRIBE,2);
2690 TANC(INICIALIZA,LEE1,0);
2700 TANC(INICIALIZA,LEE1,1);
2710 TANC(INICIALIZA,LEE1,2);
2720 TANC(INICIALIZA,LEE1,ESCRIBE,0);

```

```

00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002165
00002166
00002168
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320
00002330
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720

```


2740	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 1));	00002730
2750	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 2));	00002740
2760	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 3));	00002750
2770	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 4));	00002760
2780	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 5));	00002770
2790	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 6));	00002780
2800	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 7));	00002790
2810	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 8));	00002800
2820	SSC(INICIALI ZA, LEEF, ESCRIBIBE, 9));	00002810
2830	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 0));	00002820
2840	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 1));	00002830
2850	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 2));	00002840
2860	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 3));	00002850
2870	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 4));	00002860
2880	SETC(TNICIALI ZA, LEEF, ESCRIBIBE, 5));	00002870
2890	TASC(TNICIALI ZA, LEEF1, ESCRIBIBE, 6));	00002880
2900	TASC(TNICIALI ZA, LEEF1, ESCRIBIBE, 7));	00002890
2910	TASC(TNICIALI ZA, LEEF1, ESCRIBIBE, 8));	00002900
2920	GRAC(TNICIALI ZA, LEEF, 1));	00002910
2930	GRAC(TNICIALI ZA, LEEF, 2));	00002920
2940	GRAC(TNICIALI ZA, LEEF, 3));	00002930
2950	GRAC(TNICIALI ZA, LEEF, 4));	00002940
2960	END;	00002950
2970	I:=ANN+2;	00002960
2980	END	00002970
2990	ELSE IF I=NNN	00002980
3000	THEN BEGIN	00002990
3010	WRITE(SAL, <*** ERROR SINTACTICO ***>);	00003000
3020	WRITE(SAL, <***>);	00003010
3030	END;	00003020
3040	END;	00003030
3050	GO DTRC;	00003040
3060	END;	00003050
3070	END;	00003060
3100	AFUERA:	00003070
3200	END.	00003100
		00003200

```

70      Z
80      Z SECCION 2.3.4 INCISO 1.
90      Z
100     $ SET LEVEL 3
110     $ SET AUTOBIND
120     $ HOST IS OBJECT/SIMASET
200     I
300     $LOADINFO "ISAL/SIMASET/INFO"
400     PROCEDURE MAX;EXTERNAL;
500     PROCEDURE MIN;EXTERNAL;
600     PROCEDURE MEDIA;EXTERNAL;
700     PROCEDURE NORMA2;EXTERNAL;
800     PROCEDURE NORMINF;EXTERNAL;
900     PROCEDURE NORMA1;EXTERNAL;
1000    PROCEDURE DESVIA;EXTERNAL;
1100    I
1200    PROCEDURE SEN(INICIALIZA,F,Q);
1300    PROCEDURE INICIALIZA;
1400    PROCEDURE F(X1,X2);
1500    FILE X1,X2;
1600    FORMAL;
1700    REAL Q;
1800    BEGIN
1900    LABEL L2;
2000    REAL I,J;
2100    PROCEDURE MENSAJE(K);
2200    REAL K;
2300    BEGIN
2400    IF K=0 THEN
2500    WRITE(SAL," * EL COMANDO OBTIENE EL MAXIMO",/,
2600    " VALOR GLOBAL DE SUS DATOS * ">);
2700    IF K=1 THEN
2800    WRITE(SAL," * EL COMANDO OBTIENE EL MINIMO",/,
2900    " VALOR GLOBAL DE SUS DATOS * ">);
3000    IF K=2 THEN
3100    WRITE(SAL," * EL COMANDO OBTIENE LA MEDIA DE LOS",/,
3200    " DATOS ALMACENADOS EN DISCO * ">);
3300    IF K=3 THEN
3400    WRITE(SAL," * EL COMANDO OBTIENE LA NORMA 2 * ">);
3500    IF K=4 THEN
3600    WRITE(SAL," * EL COMANDO OBTIENE LA NORMA INFINITO * ">);
3700    IF K=5 THEN
3800    WRITE(SAL," * EL COMANDO OBTIENE LA NORMA 1 * ">);
3900    IF K=6 THEN
4000    WRITE(SAL," * EL COMANDO OBTIENE LA DESVIACION STANDARD * ">);
4100    WRITE(SAL," * EJECUTA PRIMERO EL COMANDO LEESERIE * ">);
4200    END;
4300    PROCEDURE CHEGUEA(F);
4400    PROCEDURE F;
4500    IF FDCO=2
4600    THEN
4700    GO L2
4800    ELSE
4900    F;
5000    INICIALIZA;
5100    CASE Q OF
5200    BEGIN
5300    BEGIN
5400    MENSAJE(Q);
5500    FDCO:=1;
5600    F(CARCHI,DIRECTORIO);
5700    CHEGUEA(MAX);
5800    END;

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800

```

5900	BEGIN	00005900
6000	MENSAJE(1);	00006000
6100	FOCO:=1;	00006100
6200	F(ARCHI, DIRECTORIO);	00006200
6300	CHEQUEA(MIN);	00006300
6400	END;	00006400
6500	BEGIN	00006500
6600	MENSAJE(2);	00006600
6700	FOCO:=1;	00006700
6800	F(ARCHI, DIRECTORIO);	00006800
6900	CHEQUEA(MEDIA);	00006900
7000	END;	00007000
7100	BEGIN	00007100
7200	MENSAJE(3);	00007200
7300	FOCO:=1;	00007300
7400	F(ARCHI, DIRECTORIO);	00007400
7500	CHEQUEA(NORMA2);	00007500
7600	END;	00007600
7700	BEGIN	00007700
7800	MENSAJE(4);	00007800
7900	FOCO:=1;	00007900
8000	F(ARCHI, DIRECTORIO);	00008000
8100	CHEQUEA(NORMA1);	00008100
8200	END;	00008200
8300	BEGIN	00008300
8400	MENSAJE(5);	00008400
8500	FOCO:=1;	00008500
8600	F(ARCHI, DIRECTORIO);	00008600
8700	CHEQUEA(NORMA1);	00008700
8800	END;	00008800
8900	BEGIN	00008900
9000	MENSAJE(6);	00009000
9100	FOCO:=1;	00009100
9200	F(ARCHI, DIRECTORIO);	00009200
9300	CHEQUEA(DESVIA);	00009300
9400	END;	00009400
9500	WRITE(SAL, <<"RN", />>);	00009500
9600	L2:	00009600
9700	END.	00009700
9800		00009800

70	Z		00000070
80	Z	SECCION 2.3.4.1 COMANDO SEN/1(MAX) ABBREVIACION SN1.	00000080
90	Z		00000090
100	%	SET LEVEL 3	00000100
110	%	SET AUTOBIND	00000110
120	%	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	%	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE MAX;	00000500
600		BEGIN	00000600
700		REAL I,J,MAX1;	00000700
800		FORMAT F10X10,"DATO [",I4,"]",F20.5);	00000800
900		EBCDIC APPAY FORMATO(10);	00000900
1000		MAX1:=AAAI1; I:=1;	00001000
1100		FOR I:=2 STEP 1 UNTIL TAM DO	00001100
1200		IF AAAI1 > MAX1	00001200
1300		THEN BEGIN	00001300
1400		MAX1:=AAAI1;	00001400
1500		J:=I;	00001500
1600		END;	00001600
1700		WRITE(SAL,<<30(" ")>>);	00001700
1800		WRITE(SAL,<<"/, "EL VALOR MAXIMO ES">>);	00001800
1900		REPLACE FORMATO BY "DATO [",J FOR " DIGITS,"I";	00001900
2000		WRITE(SAL,<<X10,A11,F20.5>>,FORMATO,MAX1);	00002000
2100		WRITE(SAL,<<30(" ")>>);	00002100
2200		END.	00002200

70	Z		00000070
80	Z	SECCION 2.3.4.2 COMANDO SEN/2(MIN) ABBREVIACION SN2.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AJTDBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	S	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE MIN:	00000500
600		BEGIN	00000600
700		REAL I,J,MINI;	00000700
800		EBCDIC ARRAY FORMAT(0:10);	00000800
900		MINI:=AAA(1); I:=1;	00000900
1000		FOR I:=2 STEP 1 UNTIL TAM DO	00001000
1100		IF AAA(I) < MINI	00001100
1200		THEN BEGIN	00001200
1300		MINI:=AAA(I);	00001300
1400		J:=I;	00001400
1500		END;	00001500
1600		WRITE(SAL,<<30(" ")>>);	00001600
1700		WRITE(SAL,<<"/, "EL VALOR MINIMO ES">>);	00001700
1800		REPLACE FORMAT BY "DATA (",J FOR " DIGITS,"";	00001800
1900		WRITE(SAL,<<X10.A1,F20.5>>,FORMATO,MINI);	00001900
2000		WRITE(SAL,<<30(" ")>>);	00002000
2100		END.	00002100

70	Z		00000070
80	Z	SECCION 2.3.4.3 COMANDO SEN/3(MEDIA) ABREVIACION SN3.	00000080
90	Z		00000090
100	\$	SET LEVEL 3	00000100
110	\$	SET AUTOBIND	00000110
120	\$	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	\$	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE MEDIA;	00000500
600		BEGIN	00000600
700		REAL I,J,MED1;	00000700
800		FOR I:=1 STEP 1 UNTIL TAM DG	00000800
900		MED1:=**AAAT I);	00000900
1000		IF TAM NEG 0 THEN	00010000
1100		MED1:=MED1/TAM;	00010100
1200		WRITE(SAL,<30(" ")>);	00010200
1300		WRITE(SAL,<"/ "LA MEDIA DE SUS DATOS ES">);	00010300
1400		WRITE(SAL,<"X15,F20.5">,MED1);	00010400
1500		WRITE(SAL,<30(" ")>);	00010500
1600		END.	00010600

70	X		00000070
80	Z	SECCION 2.3.4.4 COMANDO SEN/4(NORM2) ABREVIACION SN4.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	E		00000200
300	S	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE NORMA2;	00000500
600		BEGIN	00000600
700		REAL AUX,T,NORM2;	00000700
800		AUX:=0;	00000800
900		FOR I:=1 STEP 1 UNTIL TAM DO	00000900
1000		AUX:=**((AAAE:1)**2);	00001000
1100		NORM2:=SQRT(AUX);	00001100
1200		WRITE(SAL,<30(" ")>);	00001200
1300		WRITE(SAL,<"/X2, "LA NORMA 2 DE LA SERIE",/,	00001300
1400		X10,"ES",F20.5,/,NORM2);	00001400
1500		WRITE(SAL,<30(" ")>);	00001500
1600		END.	00001600

70	Z	SECCION 2.3.4.5 COMANDO SEN/5(NORMINF) ABEVIACION SNS.	00000070
80	Z		00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	E		00000200
300	E	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	J		00000400
500		PROCEDURE NORMINF;	00000500
600		BEGIN	00000600
700		REAL I,MAXABS;	00000700
800		MAXABS:=0;	00000800
900		FOR I:=1 STEP 1 UNTIL TAM DC	00000900
1000		IF MAXABS < ABS(AAII)	00001000
1100		THEN	00001100
1200		MAXABS:=ABS(AAII);	00001200
1300		WRITE(SAL,<30(" ")>);	00001300
1400		WRITE(SAL,<"/X2>"LA NORMA INFINITO DE LA SERIE"/,	00001400
1500		X10>"S",F20.5,/>,MAXABS);	00001500
1600		WRITE(SAL,<30(" ")>);	00001600
1700		END.	00001700

70	Z		00000070
80	Z	SECCION 2.3.4.6 COMANDO SEN/6(NORM1) ABREVIACION SN6.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	I		00000200
300	S	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE NORMAL:	00000500
600		BEGIN	00000600
700		REAL AUX,T,NORM1;	00000700
800		AUX:=0;	00000800
900		FOR I:=1 STEP 1 UNTIL TAM DO	00000900
1000		AUX:==+AFS(AAA(I));	00001000
1100		NORM1:=AUX;	00001100
1200		WRITE(SAL,<<30(" ")>>);	00001200
1300		WRITE(SAL,<</>X2,"LA NORMA 1 DE LA SERIE",/);	00001300
1400		X10,"ES",F20.5,/>>,NORM1);	00001400
1500		WRITE(SAL,<<30(" ")>>);	00001500
1600		END.	00001600

70	X		00000070.
80	X	SECCION 2.3.4.7 COMANDO SEN/7(DESVIA) ABREVIACION SN7.	00000080
90	X		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	[LOADINFO =ISAL/SIMASET/INFO	00000300
400]		00000400
500		PROCEDURE DESVIA;	00000500
600		BEGIN	00000600
700		REAL I,J,MED1,AUX,DESVIA1;	00000700
800		MED1:=0;AUX:=0;	00000800
900		FOR I:=1 STEP 1 UNTIL TAM DO	00000900
1000		MED1:=**+AAAIJ;	00001000
1100		IF TAM NEC 0 THEN	00001100
1200		MED1:=MED1/TAM;	00001200
1300		FOR I:=1 STEP 1 UNTIL TAM DO	00001300
1400		AUX:=**+(AAAIJ)-MED1)**2;	00001400
1500		IF TAM NEC 0 THEN	00001500
1600		DESVIA1:=SQRT(AUX/TAM);	00001600
1700		WRITE(SAL,<30(" ")>);	00001700
1800		WRITE(SAL,<"/, "LA DESVIACION STANDARD DE SUS DATOS ES">);	00001800
1900		WRITE(SAL,<X15,F20.5>,DESVIA1);	00001900
2000		WRITE(SAL,<30(" ")>);	00002000
2100		END.	00002100

```

70      Z SECCION 2.3.4 TNCISO 2.
80      Z
90      Z
100     $ SET LEVEL 3
110     $ SET AUTOSIND
120     $ HOST IS OBJECT/SIMASET
200     L
300     $LOADINFO "ISAL/SIMASET/INFO"
400     PROCEDURE SUMA(TAM1,TAM2,A1,B1);REAL TAM1,TAM2;ARRAY A1,B1(1);EXTERNAL;
500     PROCEDURE MULT(TAM1,TAM2,A1,B1);REAL TAM1,TAM2;ARRAY A1,B1(1);EXTERNAL;
600     PROCEDURE CONVOL(TAM1,TAM2,A1,B1);REAL TAM1,TAM2;ARRAY A1,B1(1);
700         EXTERNAL;
800     ]
900     PROCEDURE SES(INICIALIZA,F,G,S);
1000    PROCEDURE INICIALIZA;
1100    PROCEDURE F(X1,X2);
1200        FILE X1,X2;
1300    FORMAL;
1400    PROCEDURE G(Q,X1,X2);
1500        REAL Q;
1600        FILE X1,X2;
1700    FORMAL;
1800    REAL S;
1900    BEGIN
2000    LABEL L2;
2100    REAL I,TAM1,TAM2;
2200    ARRAY A1,B1(1);HC(0);
2300    PROCEDURE SFNAFCRCC(X,XX);
2400        REAL X;
2500        ARRAY XX(1);
2600        IF FCCD=2
2700        THEN
2800            GO L2
2900        ELSE
3000            IF FCCD=1
3100            THEN BEGIN
3200                X=TAM;
3300                RESIZE(XX,X);
3400                FOR I:=1 STEP 1 UNTIL X DO
3500                    XX(I):=AAAI;
3600            END;
3700    PROCEDURE MENSAJE(K);
3800        REAL K;
3900        BEGIN
4000            IF K=0 THEN
4100                WRITE(SAL,<"* EL COMANDO SUMA LOS VALORES DE",/,
4200                    " LAS SERIES ALMACENADAS EN DISCO *">);
4300            IF K=1 THEN
4400                WRITE(SAL,<"* EL COMANDO MULTIPLICA LOS VALORES DE",/,
4500                    " LAS SERIES ALMACENADAS EN DISCO *">);
4600            IF K=2 THEN
4700                WRITE(SAL,<"* EL COMANDO REALIZA LA CONVOLUCION",/,
4800                    " DE LAS SERIES ALMACENADAS EN DISCO *">);
4900            WRITE(SAL,<"* EJECUTA DOS VECES EL COMANDO",/,
5000                    " LEESERIE *">);
5100        END;
5200    INICIALIZA;
5300    CASE S OF
5400    BEGIN
5500    BEGIN
5600        MENSAJE(0);
5700        FCCD:=1;
5800        F(ARCHI,DIRECTORIO);

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800

```

5900	SEMAFORO(TAM1,A1);	0005900
6000	F(ARCH1,DIRECTORIO);	0006000
6100	SEMAFORO(TAM2,B1);	0006100
6200	SUMACTAM1,TAM2,A1,B1);	0006200
6300	END;	0006300
6400	BEGIN	0006400
6500	MENSAJE(1);	0006500
6600	FOCO:=1;	0006600
6700	F(ARCH1,DIRECTORIO);	0006700
6800	SEMAFORO(TAM1,A1);	0006800
6900	F(ARCH1,DIRECTORIO);	0006900
7000	SEMAFORO(TAM2,B1);	0007000
7100	MULT(TAM1,TAM2,A1,B1);	0007100
7200	END;	0007200
7300	BEGIN	0007300
7400	MENSAJE(2);	0007400
7500	FOCO:=1;	0007500
7600	F(ARCH1,DIRECTORIO);	0007600
7700	SEMAFORO(TAM1,A1);	0007700
7800	F(ARCH1,DIRECTORIO);	0007800
7900	SEMAFORO(TAM2,B1);	0007900
8000	CONVOL(TAM1,TAM2,A1,B1);	0008000
8100	END;	0008100
8200	END;	0008200
8300	WRITE(SAL, <"* DESEA ALMACENAR SUS NUEVOS VALORES *">);	0008300
8400	READ(ENT, <A?>, HCO);	0008400
8500	IF HCO="SI"	0008500
8600	THEN BEGIN	0008600
8700	FOCO:=1;	0008700
8800	GOTO ARCH1,DIRECTORIO);	0008800
8900	END;	0008900
9000	WRITE(SAL, <"RN"/>);	0009000
9100	L2:	0009100
9200	END.	0009200

```

70      Z
80      Z SECCION 2.3.4.2 COMANDO SES/1(SUMA) ABREVIACION SE1.
90      Z
100     S SET LEVEL 3
110     S SET AUTODIAG
120     S HOST IS OBJECT/SIMASET
200     [
300     SLOADINFO "ISAL/SIMASET/INFO"
500     ]
600     PROCEDURE SUMA(TAM1,TAM2,A1,B1);
610     REAL TAM1,TAM2;
620     ARRAY A1,B1(1);
700     BEGIN
800     REAL I,AUX,C,D;
810     EOC DIC ARRAY FORMAT(0:10);
820     LABEL LL;
900     IF TAM1 LEQ TAM2
1000    THEN BEGIN
1100    TAM:=TAM2;
1200    RESIZE(CCC,TAM2);
1300    FOR I:=1 STEP 1 UNTIL TAM1 DO
1400    CCC(I):=A1(I)+B1(I);
1500    FOR I:=TAM1+1 STEP 1 UNTIL TAM2 DO
1600    CCC(I):=B1(I);
1700    END;
1800    IF TAM1 > TAM2
1900    THEN BEGIN
2000    TAM:=TAM1;
2100    RESIZE(CCC,TAM1);
2200    FOR I:=1 STEP 1 UNTIL TAM2 DO
2300    CCC(I):=A1(I)+B1(I);
2400    FOR I:=TAM2+1 STEP 1 UNTIL TAM1 DO
2500    CCC(I):=A1(I);
2600    END;
2700    AUX:=MAX(TAM1,TAM2);
2800    WRITE(SAL,<"* DESEA CONOCER SUS RESULTADOS *">);
2900    READ(ENT,<A2>,HEC);
3000    IF HEC="SI"
3100    THEN BEGIN
3200    LL:
3300    WRITE(SAL,<"* ESPECIFIQUE EL INTERVALO *">);
3400    READ(ENT,/,C,D);
3500    IF C < 1 OR D > AUX
3600    THEN BEGIN
3700    WRITE(SAL,<"* ERROR EN EL INTERVALO *">);
3800    WRITE(SAL,<X10,"EL TAMANO DE LA SERIE VARIA DE 1 A",I6>,AUX);
3900    GO LL;
4000    END
4100    ELSE
4200    FOR I:=C STEP 1 UNTIL D DO
4300    BEGIN
4400    REPLACE FORMAT BY " " FOR 11;
4500    REPLACE FORMAT BY "D0" FOR * DIGITS,"";
4600    WRITE(SAL,<X10,A1,F20.5>,FORMAT,CCC(I));
4700    END;
4800    END;
4900    END.

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000500
00000600
00000610
00000620
00000700
00000800
00000810
00000820
00000900
00010000
00011000
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00020000
00021000
00022000
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00030000
00031000
00032000
00033000
00033400
00033500
00033600
00033700
00033800
00033900
00040000
00040100
00040200
00040300
00040400
00040500
00040600
00040700
00040800
00040900

```

```

70      Z
80      Z SECCION 2.3.4.9 COMANDO SES/2(MULT) ABREVIACION SE2.
90      Z
100     S SET LEVEL 3
110     S SET AUTOBIND
120     S HOST IS OBJECT/SIMASET
200     C
300     SLOADINFO "ISAL/SIMASET/INFO"
500     I
600     PROCEDURE MULT(TAM1,TAM2,A1,B1);
610     REAL TAM1,TAM2;
620     ARRAY A1,B1(1);
700     BEGIN
800     REAL I,AUX,C,D;
810     EBCDIC ARRAY FORMAT(0:10);
820     LABEL LL;
900     IF TAM1 LEQ TAM2
1000    THEN BEGIN
1100     TAM:=TAM2;
1200     RESIZE(CCC,TAM2);
1300     FOR I:=1 STEP 1 UNTIL TAM1 DO
1400     CCC(I):=A1(I)*B1(I);
1500     FOR I:=TAM1+1 STEP 1 UNTIL TAM2 DO
1600     CCC(I):=B1(I);
1700     END;
1800     IF TAM1 > TAM2
1900     THEN BEGIN
2000     TAM:=TAM1;
2100     RESIZE(CCC,TAM1);
2200     FOR I:=1 STEP 1 UNTIL TAM2 DO
2300     CCC(I):=A1(I)*B1(I);
2400     FOR I:=TAM2+1 STEP 1 UNTIL TAM1 DO
2500     CCC(I):=A1(I);
2600     END;
2700     AUX:=MAX(TAM1,TAM2);
2800     WRITE(SAL,<"* DESEA CONOCER SUS RESULTADOS *">);
2900     READ(ENT,<A2>,HI(1));
3000     IF HI(1)="SI"
3100     THEN BEGIN
3200     LL:
3300     WRITE(SAL,<"* ESPECIFIQUE EL INTERVALO *">);
3400     READ(ENT,/,C,D);
3500     IF C < 1 OR D > AUX
3600     THEN BEGIN
3700     WRITE(SAL,<"** ERROR EN EL INTERVALO **">);
3800     WRITE(SAL,<X1>,"EL TAMAÑO DE LA SERIE VARIA DE 1 A ",I6>,AUX);
3900     GO LL;
4000     END
4100     ELSE
4200     FOR I:=C STEP 1 UNTIL D DO
4300     BEGIN
4400     REPLACE FORMATO BY " " FOR 1;
4500     REPLACE FORMATO BY "DATO [",] FOR * DIGITS,"";
4600     WRITE(SAL,<X1>,A1,F20.5>,FORMATO,CCC(I));
4700     END;
4800     END;
4900     END.

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000500
00000600
00000610
00000620
00000700
00000800
00000810
00000820
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900

```

70	2		00000070
80	2	SECCION 2.3.4.10 COMANDO SES/3(CONVOL) ABBREVIACION SE3.	00000080
90	2		00000090
100	2	SET LEVEL 3	00000100
110	2	SET AUTOBIND	00000110
120	2	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	SLOADINFO "ISAL/SIMASET/INFO"		00000300
400]		00000400
500		PROCEDURE CONVCL(TAM1,TAM2,A1,B1);	00000500
600		REAL TAM1,TAM2;	00000500
700		ARRAY A1,B1(1);	00000600
800		BEGIN	00000800
900		WRITE(SAL, <<"* OBSERVACION :"/,	00000900
1000		X10,"ALGORITMO DE CONVOLUCION NO IMPLEMENTADO"/,	00001000
1100		X10,"EL PROGRAMA OR DEBE TOMAR EN CUENTA QUE LA"/,	00001100
1200		X10,"RUTINA OPERA CON 2 SERIES, LA UNA ESTA EN"/,	00001200
1300		X10,"EL ARREGLO A(1:1:TAM1), LA OTRA EN EL ARRE-"/,	00001300
1400		X10,"GLO P(1:1:TAM2), ASI MISMO LOS RESULTADOS"/,	00001400
1500		X10,"OBTENIDOS DEBEN SER GUARDADOS EN EL ARRE-"/,	00001500
1600		X10,"GLO C(1:1) UNDE I DEBE VARIAR DE 1 AL"/,	00001600
1700		X10,"MAX(TAM1,TAM2). *"/>);	00001700
1800		END.	00001800

```

70      Z
80      Z SECCION 2.3.4 INCISO 3.
90      Z
100     $ SET LEVEL 3
110     $ SET AUTOBIND
120     $ HOST IS OBJECT/SIMASET
200     [
300     $LOADINFO "ISAL/SIMASET/INFO"
400     PROCEDURE SUMAS(TAM1,TAM2,A1,B1);REAL TAM1,TAM2;
500     ARRAY A1,B1(1);EXTERNAL;
600     PROCEDURE PREDICTOR(TAM1,TAM2,ORDEN,A1,B1);REAL TAM1,TAM2,ORDEN;
700     ARRAY A1,B1(1);EXTERNAL;
800     PROCEDURE CONTROLA(TAM1,TAM2,A1,B1);REAL TAM1,TAM2;
900     ARRAY A1,B1(1);EXTERNAL;
1000    ]
1100    PROCEDURE TAT(INICIALIZA,F,G,S);
1200    PROCEDURE INICIALIZA;
1300    PROCEDURE F(X1,X2);
1400    FILE X1,X2;
1500    FORMAL;
1600    PROCEDURE G(Q,X1,X2);
1700    REAL Q;
1800    FILE X1,X2;
1900    FORMAL;
2000    REAL S;
2100    BEGIN
2200    LABEL L2;
2300    REAL I,TAM1,TAM2,ORDEN,N1,N2,M1,M2,P1,P2;
2400    ARRAY A1,B1(0:0),FOC(0);
2500    PROCEDURE SFMAFORC(X,XX);
2600    REAL X;
2700    ARRAY XX(1);
2800    IF FOC=2
2900    THEN
3000    GO L2
3100    ELSE
3200    IF FOC=1
3300    THEN BEGIN
3400    X:=TAM;
3500    RESIZE(XX,X);
3600    FOR I:=1 STEP 1 UNTIL X DO
3700    XX(I):=AAAI;
3800    END;
3900    PROCEDURE MENSAJE(K);
4000    REAL K;
4100    BEGIN
4200    IF K=0 THEN
4300    WRITE(SAL,<<" EL COMANDO SUMA LOS VALORES DE"/,
4400    " LAS TABLAS ALMACENADAS EN DISCO *">>);
4500    IF K=1 THEN
4600    WRITE(SAL,<<" EL COMANDO OBTIENE LOS COEFICIENTES FS Y US DEL"/,
4700    " X20,PREDICTOR. *">>);
4800    IF K=2 THEN
4900    WRITE(SAL,<<" EL COMANDO REALIZA EL CONTROLADOR *">>);
5000    WRITE(SAL,<<" EJECUTA EL COMANDO LEETABLA *">>);
5100    END;
5200    INICIALIZA;
5300    A1(0):=1; B1(0):=1;
5400    CASE S OF
5500    BEGIN
5600    MENSAJE(0);
5700    FOC:=1;

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700

```



```

5800 F(TABLA,DIRTAB);
5900 N1:=N; M1:=M; P1:=P;
6000 SEMAFORO(TAM1,A1);
6100 F(TABLA,DIRTAB);
6200 N2:=N; M2:=M; P2:=P;
6300 IF M1=M2 AND P1=P2
6400 THEN BEGIN
6500 SEMAFORO(TAM2,B1);
6600 SUMAS(TAM1,TAM2,A1,B1);
6700 END;
6800 ELSE BEGIN
6900 WRITE(SAL, <<" ERROR EN LAS DIMENSIONES DE LAS TABLAS *">>);
7000 WRITE(SAL, <<"R", />);
7100 GO L2;
7200 END;
7300 END;
7400 BEGIN
7500 MENSAJE(1);
7600 FOCO:=1;
7700 F(TABLA,DIRTAB);
7800 N1:=N; M1:=M; P1:=P;
7900 SEMAFORO(TAM1,A1);
8000 F(TABLA,DIRTAB);
8100 N2:=N; M2:=M; P2:=P;
8200 IF M1=1 AND P1=1 AND M2=1 AND P2=1
8300 THEN BEGIN
8400 SEMAFORO(TAM2,B1);
8500 PREDICTOR(TAM1,TAM2,ORDEN,A1,B1);
8600 END;
8700 ELSE BEGIN
8800 WRITE(SAL, <<" NO SE PUEDEN CALCULAR LAS FS Y US DEL ", /,
8900 "X15, "PREDICTOR . *">>);
9000 WRITE(SAL, <<"R", />);
9100 GO L2;
9200 END;
9300 END;
9400 BEGIN
9500 MENSAJE(2);
9600 FOCO:=1;
9700 F(TABLA,DIRTAB);
9800 N1:=N; M1:=M; P1:=P;
9900 SEMAFORO(TAM1,A1);
10000 F(TABLA,DIRTAB);
10100 N2:=N; M2:=M; P2:=P;
10200 WRITE(SAL, <<" OBSERVACION 1 : ", /, X10, "EL PROGRAMADOR",
10300 " DEBE DEFINIR LAS CONDICIONES INICIALES NECESARIAS", /,
10400 X10, "EN EL ALGORITMO PADRE ALG/TAT, PARA ESTO PUEDE", /,
10500 X10, "HACER UN LISTADO DE LA 7400-9200 DE ALG/TAT, QUE", /,
10600 X10, "SE REFIERE A COMO SE IMPLEMENTO PARA EL CASO DEL", /,
10700 X10, "PREDICTOR. *">>);
10800 SEMAFORO(TAM2,B1);
10900 CONTROLA(TAM1,TAM2,A1,B1);
11000 END;
11100 END;
11200 WRITE(SAL, <<" DESEA ALMACENAR SUS NUEVOS VALORES *">>);
11300 READ(ENT, <<A2>>, HCO);
11400 IF HCO="SI"
11500 THEN BEGIN
11600 FOCO:=1;
11700 IF S=0 THEN IF N1 > N2
11800 THEN ORDEN:=N1;
11900 ELSE ORDEN:=N2;
12000 IF S=1 THEN P1:=TAM1;
12100 WRITE(SAL, <<" OBSERVACION: *">>);
12200 WRITE(SAL, <<X10, " EL ORDEN DE SU NUEVA TABLA ES ", I4, >>, ORDEN);

```

```

00005800
00005900
00006000
00006100
00006200
00006300
00006400
00006500
00006600
00006700
00006800
00006900
00007000
00007100
00007200
00007300
00007400
00007500
00007600
00007700
00007800
00007900
00008000
00008100
00008200
00008300
00008400
00008500
00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100

```

12300
12400
12500
12600
12700
12800

```
WRITE(SAL, <X10, " EL NUMERO DE FILAS ES ", I4, X1, " * > ", M1);  
GC(1, TABLA, DIRTAB);  
END;  
WRITE(SAL, <"NR" />);  
L2:  
END.
```

C0012300
C0012400
C0012500
C0012600
C0012700
C0012800

70	X		00000070
80	Z	SECCION 2.3.4.11 COMANDO TAT/1(SUMA) ABREVIACION TT1.	00000080
90	Z		00000090
100	6	SET LEVEL 3	00000100
110	8	SET AUTOBIND	00000110
120	8	HOST IS OBJECT/SIMASET	00000120
200	1		00000200
300	1	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	1		00000400
500		PROCEDURE SUMAS(TAM1,TAM2,A1,B1);	00000500
600		REAL TAM1,TAM2;	00000600
700		ARRAY A1,B1(1);	00000700
800		BEGIN	00000800
900		REAL I;	00000900
1000		IF TAM1 LEQ TAM2	00001000
1100		THEN BEGIN	00001100
1200		TAM:=TAM2;	00001200
1300		RESIZE(CCC,TAM2);	00001300
1400		FOR I:=1 STEP 1 UNTIL TAM1 DO	00001400
1500		CCC(I):=A1(I)+B1(I);	00001500
1600		FOR I:=TAM1+1 STEP 1 UNTIL TAM2 DO	00001600
1700		CCC(I):=B1(I);	00001700
1800		END;	00001800
1900		IF TAM1 > TAM2	00001900
2000		THEN BEGIN	00002000
2100		TAM:=TAM1;	00002100
2200		RESIZE(CCC,TAM1);	00002200
2300		FOR I:=1 STEP 1 UNTIL TAM2 DO	00002300
2400		CCC(I):=A1(I)+B1(I);	00002400
2500		FOR I:=TAM2+1 STEP 1 UNTIL TAM1 DO	00002500
2600		CCC(I):=A1(I);	00002600
2700		END;	00002700
2800		END.	00002800

70			00000070
80	Z	SECCION 2.3.4.12 COMANDO TAT/2(PREDICT) ABREVIACION TT2.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	LOADINFO "ISAL/SIMASET/INFO"		00000300
400]		00000400
500		PROCEDURE PREDICTOR(TAM1,TAM2,ORDEN,A1,B1);	00000500
600		REAL TAM1,TAM2,ORDEN;	00000600
700		ARRAY A1,B1(1);	00000700
800		BEGIN	00000800
900		REAL I,J,K,LL,MM,PP,SUMA,SUMA1,SUMA2,AUX;	00000900
1000		WRITE(SAL, < " * CUAL ES EL PASO K DEL PREDICTOR * " >);	00001000
1100		READ(ENT, /, K);	00001100
1200		BEGIN	00001200
1300		ARRAY FIC(1:K-1), UIC(1:TAM1-1), CEROC(TAM1-K+1:TAM1-1);	00001300
1400		FIC(1) := B1(1) - A1(1);	00001400
1500		FOR J := 2 STEP 1 UNTIL K-1 DO	00001500
1600		BEGIN	00001600
1700		SUMA := A1(J);	00001700
1800		FOR I := 2 STEP 1 UNTIL J DO	00001800
1900		SUMA := * + A1(J-I+1) * FIC(I-1);	00001900
2000		FIC(J) := B1(J) - SUMA;	00002000
2100		END;	00002100
2200		LL := 0;	00002200
2300		MM := K;	00002300
2400		FOR J := 1 STEP 1 UNTIL TAM1-K DO	00002400
2500		BEGIN	00002500
2600		LL := * + 1; PP := LL;	00002600
2700		SUMA1 := A1(MM);	00002700
2800		FOR I := 1 STEP 1 UNTIL K-1 DO	00002800
2900		BEGIN	00002900
3000		SUMA1 := * + A1(PP) * FIC(K-I);	00003000
3100		PP := * + 1;	00003100
3200		END;	00003200
3300		UIC(J) := B1(MM) - SUMA1;	00003300
3400		MM := * + 1;	00003400
3500		END;	00003500
3600		LL := 0;	00003600
3700		FOR J := TAM1-K+1 STEP 1 UNTIL TAM1-1 DO	00003700
3800		BEGIN	00003800
3900		LL := * + 1; PP := LL;	00003900
4000		SUMA2 := 0;	00004000
4100		FOR I := 1 STEP 1 UNTIL K-LL DO	00004100
4200		BEGIN	00004200
4300		SUMA2 := * + A1(TAM1-I+1) * FIC(PP);	00004300
4400		PP := * + 1;	00004400
4500		END;	00004500
4600		UIC(J) := CEROC(J) - SUMA2;	00004600
4700		END;	00004700
4800		RESIZE(CCC, TAM1+TAM1);	00004800
4900		ORDEN := 1;	00004900
5000		AUX := 1;	00005000
5100		FOR I := 1 STEP 1 UNTIL K-1 DO	00005100
5200		BEGIN	00005200
5300		CCC(AUX) := FIC(I);	00005300
5400		AUX := * + 1;	00005400
5500		END;	00005500
5600		FOR I := K STEP 1 UNTIL TAM1 DO	00005600
5700		BEGIN	00005700

5900	AUX:=**+1;	00005900
6000	END;	00006000
6100	FOR I:=1 STEP 1 UNTIL TAM1 DO	00006100
6200	BEGIN	00006200
6300	CCC[AUX]:=U[I-1];	00006300
6400	AUX:=**+1;	00006400
6500	END;	00006500
6600	WRITE(SAL, <"* DESEA CONOCER SUS RESULTADOS **>);	00006600
6700	READ(ENT, <A2>, I(0));	00006700
6800	IF HI(I) = "SI"	00006800
6900	THEN BEGIN	00006900
7000	AUX:=1;	00007000
7100	WRITE(SAL, <3>(" - ")>);	00007100
7200	WRITE(SAL, <"- OS VALORES DE F SON", />);	00007200
7300	FOR J:=1 STEP 1 UNTIL K-1 DO	00007300
7400	BEGIN	00007400
7500	WRITE(SAL, <X10, "F [", I2, "]" , F20.5>, I, CCC[AUX]);	00007500
7600	AUX:=**+1;	00007600
7700	END;	00007700
7800	AUX:=TAM1+1;	00007800
7900	WRITE(SAL, </> "LOS VALORES DE U SON", />);	00007900
8000	FOR I:=0 STEP 1 UNTIL TAM1-1 DO	00008000
8100	BEGIN	00008100
8200	WRITE(SAL, <X10, "U [", I2, "]" , F20.5>, I, CCC[AUX]);	00008200
8300	AUX:=**+1;	00008300
8400	END;	00008400
8500	WRITE(SAL, </> 30(" - ")>);	00008500
8600	END;	00008600
8700	END;	00008700
8800	END.	00008800

70	Z								
80	Z	SECCION 2.3.4.13	COMANDO TAT/3(CONTROL)	ABREVIACION	TT3.				00000070
90	Z								00000080
100	Z								00000090
110	S	SET LEVEL 3							00000100
120	S	SET AUTOBIND							00000110
130	S	HOST IS OBJECT/SIMASET							00000120
200	E								00000200
300	E	LOADINFO "ISAL/SIMASET/INFO"							00000300
400	J								00000400
500		PROCEDURE CONTROLA(TAM1,TAM2,A1,B1);							00000500
600		REAL TAM1,TAM2;							00000600
700		ARRAY A1,B1(1);							00000700
800		BEGIN							00000800
900		WRITEC(SAL, <"*OBSERVACION 2 :"/,							00000900
1000		X10,"ALGORITMO CONTROLA NO IMPLEMENTADO."/,							00001000
1100		X10,"EL PROGRAMADOR DEBE SABER QUE ESTA ACTUANDO"/,							00001100
1200		X10,"CON DOS CONJUNTOS DE TABLAS: LA UNA ESTA"/,							00001200
1300		X10,"EN EL ARREGLO A1(1:TAM1) (A1(0)=1); Y LA"/,							00001300
1400		X10,"CIRA ESTA EN B1(1:TAM2) (B1(0)=1). SI SE"/,							00001400
1500		X10,"DESEAN ALMACENAR LOS RESULTADOS DEBE USAR"/,							00001500
1600		X10,"SE EN LA RUTINA EL ARREGLO CCCCII, DONDE EL"/,							00001600
1610		X10,"TAMANO DEL ARREGLO DEBE SER ACTUALIZADO CON RESIZE,Y"/,							00001700
1700		X10,"EL INDICE I VARIA DE 1 AL MAX(TAM1,TAM2). *">);							00001800
1800		END.							

70	Z		00000070
80	Z	SECCION 2.3.4 INCISO 4.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	L		00000200
300	S	LOADINFO "ISAL/SIMASET/INFB"	00000300
400	PROCEDURE	MAXI;EXTERNAL;	00000400
500	PROCEDURE	MINI;EXTERNAL;	00000500
600	PROCEDURE	MEDI;EXTERNAL;	00000600
700	J		00000700
800	PROCEDURE	TAN(INICIALIZA,F,Q);	00000800
900	PROCEDURE	INICIALIZA;	00000900
1000	PROCEDURE	F(X1,X2);	00001000
1100	FILE	X1,X2;	00001100
1200	FORMAL;		00001200
1300	REAL	G;	00001300
1400	BEGIN		00001400
1500	LABEL	L2;	00001500
1600	REAL	I,J,MAXI,MINI,MEDI;	00001600
1700	PROCEDURE	MENSAJE(K);	00001700
1800	REAL	K;	00001800
1900	BEGIN		00001900
2000	IF	K=0 THEN	00002000
2100	WRITE(CSAL,<" * EL COMANDO OBTIENE EL MAXIMO"/,		00002100
2200	VALOR GLOBAL DE SUS DATOS * ">);		00002200
2300	IF	K=1 THEN	00002300
2400	WRITE(CSAL,<" * EL COMANDO OBTIENE EL MINIMO"/,		00002400
2500	VALOR GLOBAL DE SUS DATOS * ">);		00002500
2600	IF	K=2 THEN	00002600
2700	WRITE(CSAL,<" * EL COMANDO OBTIENE LA MEDIA DE LOS"/,		00002700
2800	DATOS ALMACENADOS EN DISCO * ">);		00002800
2900	WRITE(CSAL,<" * EJECUTA PRIMERO EL COMANDO LEETABLA * ">);		00002900
3000	END;		00003000
3100	PROCEDURE	CHEQUEA(F);	00003100
3200	PROCEDURE	F;	00003200
3300	IF	FCCC=2	00003300
3400	THEN		00003400
3500	GO	L2	00003500
3600	ELSE		00003600
3700	F;		00003700
3800	INICIALIZA;		00003800
3900	CASE	G OF	00003900
4000	BEGIN		00004000
4100	BEGIN		00004100
4200	MENSAJE(0);		00004200
4300	FCCC:=1;		00004300
4400	F(TABLA,DIRTAB);		00004400
4500	CHEQUEA(MAXI);		00004500
4600	END;		00004600
4700	BEGIN		00004700
4800	MENSAJE(1);		00004800
4900	FCCC:=1;		00004900
5000	F(TABLA,DIRTAB);		00005000
5100	CHEQUEA(MINI);		00005100
5200	END;		00005200
5300	BEGIN		00005300
5400	MENSAJE(2);		00005400
5500	MEDI:=0;		00005500
5600	FCCC:=1;		00005600
5700	F(TABLA,DIRTAB);		00005700
5800	WRITE(CSAL,<" * EJECUTA PRIMERO EL COMANDO LEETABLA * ">);		00005800

60005800
00006000
00006100
00006200
00006300

END.
L2:
WRITE(SAL, <"NN"> / >);
END.

60005800
00006000
00006100
00006200
00006300


```

70      Z
80      Z SECCION 2.3.4.14 COMANDO TAN/1(MAX) ABBREVIACION INI.
90      Z
100     $ SET LEVEL 3
110     $ SET AUTOBIND
120     $ HOST IS OBJECT/SIMASET
200     L
300     $LOADINFO "ISAL/SIMASET/INFO"
400     I
500     PROCEDURE MAXI;
600     BEGIN
700         REAL I,J,J1,JJ,MAXI,ORDEN,RENGLON,AUX;
800         REINGLON:=1;
900         ORDEN:=0;
1000        AUX:=P*M;
1100        MAXI:=AAAE11; I:=1;
1200        FOR I:=2 STEP 1 UNTIL TAM DO
1300            IF AAE11 > MAXI
1400                THEN BEGIN
1500                    MAXI:=AAAE11;
1600                    JJ:=I;
1700                END;
1800            J:=JJ;
1900            FOR J1:=1 STEP 1 UNTIL TAM DO
2000                IF J > AUX
2100                    THEN BEGIN
2200                        J:=**AUX;
2300                        ORDEN:=**+1;
2400                    END
2500                ELSE IF J < AUX THEN J1:=TAM+1;
2600            J:=JJ;
2700            FOR J1:=1 STEP 1 UNTIL TAM DO
2800                BEGIN
2900                    IF J < P
3000                        THEN BEGIN
3100                            J:=**P;
3200                            REINGLON:=**+1;
3300                        END
3400                    ELSE BEGIN
3500                        REINGLON:=REINGLON;
3600                        J1:=TAM+1;
3700                    END;
3800                    IF REINGLON = M+1 THEN
3900                        REINGLON:=1;
4000                END;
4100                JJ:=**((ORDEN)*M+REINGLON-1)*P;
4200            WRITE(SAL,*,N,P,K);
4300            WRITE(SAL,<<30(" ")>>);
4400            WRITE(SAL,<</, "EL VALOR MAXIMO ESTA EN LA MATRIZ :",/,
4500                X15,"ORDEN N",I6,/,
4600                X15,"REINGLON M",I6,/,
4700                X15,"COLUMNA P",I6,/,
4800                X15,"Y ES =",F20.5>>,ORDEN,REINGLON,JJ,MAXI);
4900            WRITE(SAL,<<30(" ")>>);
5000        END.

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000

```

```

70      Z
80      Z SECCION 2.3.4.15 COMANDO TAN/2(MIN) ABRUEVIACION TN2.
90      Z
100     S SET LEVEL 3
110     S SET AUTOBIND
120     S HOST IS OBJECT/SIMASET
200     [
300     [LOADINFO "ISAL/SIMASET/INFO"
400     ]
500     ]
600     PROCEDURE MINT;
700     BEGIN
800     REAL I,J,MINI,J1,JJ,ORDEN,RENGLON,AUX;
900     RENGLON:=1;
1000    ORDEN:=0;
1100    AUX:=P*M;
1200    MINI:=AAAI1; I:=1;
1300    FOR I:=2 STEP 1 UNTIL TAM DO
1400    IF AAI1 < MINI
1500    THEN BEGIN
1600        MINI:=AAAI1;
1700        JJ:=I;
1800    END;
1900    J:=JJ;
2000    FOR J1:=1 STEP 1 UNTIL TAM DO
2100    IF J > AUX
2200    THEN BEGIN
2300        J:=*-AUX;
2400        ORDEN:=**+1;
2500    END
2600    ELSE IF J < AUX THEN J1:=TAM+1;
2700    J:=JJ;
2800    FOR J1:=1 STEP 1 UNTIL TAM DO
2900    BEGIN
3000    IF J > P
3100    THEN BEGIN
3200        J:=*-P;
3300        RENGLON:=**+1;
3400    END
3500    ELSE BEGIN
3600        RENGLON:=RENGLON;
3700        J1:=TAM+1;
3800    END;
3900    IF RENGLON = M+1 THEN
4000    RENGLON:=1;
4100    END;
4200    JJ:=**-(ORDEN)*M+RENGLON-1)*P;
4300    WRITE(SAL,*,N,P,M);
4400    WRITE(SAL,<30(" ")>);
4500    WRITE(SAL,<"/,"EL VALOR MINIMO ESTA EN LA MATRIZ :"/,
4600           X15,"ORDEN N",I6,/,
4700           X15,"RENGLON M",I6,/,
4800           X15,"COLUMNA P",I6,/,
4900           X15,"Y ES =",F20.5>,ORDEN,RENGLON,JJ,MINI);
5000    END.

```

```

00000073
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000

```

70	Z		00000070
80	Z	SECCION 2.3.4.16 COMANDO TAN/3(MEDIA) ABREVIACION TN3.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIAD	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300]	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE MEDT;	00000500
600		BEGIN	00000600
700		REAL I,J,MEDI;	00000700
800		FOR I:=1 STEP 1 UNTIL TAN DO	00000800
900		MEDI:="+AADI";	00000900
1000		IF TAN NEQ 0 THEN	00001000
1100		MEDI:=MEDI/TAN;	00001100
1200		WRITE(SAL,<<30(" ")>>);	00001200
1300		WRITE(SAL,<<"/, "LA MEDIA DE SUS DATOS ES">>);	00001300
1400		WRITE(SAL,<<X15,F21.5>>,MEDI);	00001400
1500		WRITE(SAL,<<30(" ")>>);	00001500
1600		END.	00001600

```

70      Z
80      Z SECCION 2.3.4 INCISO 5.
90      Z
100     %
110     % SET LEVEL 3
120     % SET AUTOBAND
130     % HOST IS OBJECT/SIMASET
140     [
150     [LOADINFO "ISAL/SIMASET/INFO"
160     PROCEDURE FILTRA1;EXTERNAL;
170     PROCEDURE FILTRA2;EXTERNAL;
180     PROCEDURE FILTRA3;EXTERNAL;
190     PROCEDURE ESPECTRO1;EXTERNAL;
200     PROCEDURE ESPECTRO2;EXTERNAL;
210     PROCEDURE ESPECTRO3;EXTERNAL;
220     PROCEDURE SUAVIZA1;EXTERNAL;
230     PROCEDURE SUAVIZA2;EXTERNAL;
240     PROCEDURE SUAVIZA3;EXTERNAL;
250     PROCEDURE ESCALAR;EXTERNAL;
260     ]
270     ]
280     PROCEDURE SSC(INICIALIZA,F,G,S);
290     PROCEDURE INICIALIZA;
300     PROCEDURE F(X1,X2);
310     FILE X1,X2;
320     FORMAL;
330     PROCEDURE G(G,X1,X2);
340     REAL G;
350     FILE X1,X2;
360     FORMAL;
370     REAL S;
380     BEGIN
390     LABEL L2;
400     ARRAY HEO:0;
410     PROCEDURE MENSAJE(K);
420     REAL K;
430     BEGIN
440     IF K=0 THEN
450     WRITE(SAL, <<" EL COMANDO FILTRA LA SERIE *">>);
460     IF K=1 THEN
470     WRITE(SAL, <<" EL COMANDO OBTIENE EL ESPECTRO *">>);
480     IF K=2 THEN
490     WRITE(SAL, <<" EL COMANDO SUAVIZA LA SERIE *">>);
500     IF K=3 THEN
510     WRITE(SAL, <<" EL COMANDO AMPLIFICA O SIMPLIFICA A LOS",
520     " ELEMENTOS DE LA SERIE. *">>);
530     WRITE(SAL, <<" EJECUTA PRIMERO EL COMANDO LEESERIE *">>);
540     END;
550     PROCEDURE CHEQUEA;
560     IF FOCO=2
570     THEN GO L2;
580     INICIALIZA;
590     CASE S OF
600     BEGIN
610     MENSAJE(0);
620     FOCO:=1;
630     F(ARCH1.DIRECTORIO);
640     CHEQUEA;
650     FILTRA1;
660     END;
670     BEGIN
680     MENSAJE(0);
690     FOCO:=1;
700     F(ARCH1.DIRECTORIO);

```

```

00000070
00000080
00000090
00000100
00000110
00000120
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00010000
00011000
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003510
00003520
00003530
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300

```

5500	CHEQUEA;	0005500
5600	FILTRA2;	0005600
5700	END;	0005700
5800	BEGIN	0005800
5900	MENSAJE(0);	0005900
6000	FOCC:=1;	0006000
6100	F(CARCH1,DIRECTORIO);	0006100
6200	CHEQUEA;	0006200
6300	FILTRA3;	0006300
6400	END;	0006400
6500	BEGIN	0006500
6600	MENSAJE(1);	0006600
6700	FOCC:=1;	0006700
6800	F(CARCH1,DIRECTORIO);	0006800
6900	CHEQUEA;	0006900
7000	ESPECTRO1;	0007000
7100	END;	0007100
7200	BEGIN	0007200
7300	MENSAJE(1);	0007300
7400	FOCC:=1;	0007400
7500	F(CARCH1,DIRECTORIO);	0007500
7600	CHEQUEA;	0007600
7700	ESPECTRO2;	0007700
7800	END;	0007800
7900	BEGIN	0007900
8000	MENSAJE(1);	0008000
8100	FOCC:=1;	0008100
8200	F(CARCH1,DIRECTORIO);	0008200
8300	CHEQUEA;	0008300
8400	ESPECTRO3;	0008400
8500	END;	0008500
8600	BEGIN	0008600
8700	MENSAJE(2);	0008700
8800	FOCC:=1;	0008800
8900	F(CARCH1,DIRECTORIO);	0008900
9000	CHEQUEA;	0009000
9100	SUAVIZA1;	0009100
9200	END;	0009200
9300	BEGIN	0009300
9400	MENSAJE(2);	0009400
9500	FOCC:=1;	0009500
9600	F(CARCH1,DIRECTORIO);	0009600
9700	CHEQUEA;	0009700
9800	SUAVIZA2;	0009800
9900	END;	0009900
10000	BEGIN	0010000
10100	MENSAJE(2);	0010100
10200	FOCC:=1;	0010200
10300	F(CARCH1,DIRECTORIO);	0010300
10400	SUAVIZA3;	0010400
10500	END;	0010500
10510	BEGIN	0010510
10520	MENSAJE(3);	0010520
10530	FOCC:=1;	0010530
10540	F(CARCH1,DIRECTORIO);	0010540
10550	CHEQUEA;	0010550
10560	ESCALAR;	0010560
10570	END;	0010570
10600	END;	0010600
10700	WRITE(SAL,<*" DESEA ALMACENAR LA NUEVA SERIE *">);	0010700
10800	READCENT(<A2>,H(0));	0010800
10900	IF H(0)="SI"	0010900
11000	THEN BEGIN	0011000
11100	FOCC:=1;	0011100
11200	G(0,ARCH1,DIRECTORIO);	0011200
11300	END;	0011300

0011500
0011000
0011000

0011500
0011000
0011000
END

0011500
0011000
0011000

70	Z		00000070
80	Z	SECCION 2.3.4.17 COMANDO SS/1(FILTRA) ABREVIACION S11.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	I		00000200
300	I	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE FILTRAI;	00000500
600		BEGIN	00000600
700		WRITE(SAL, <<"* OBSERVACION:">>"/,	00000700
800		X10,"ALGORITMO FILTRAI NO IMPLEMENTADO."/,	00000800
900		X10,"EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	00000900
1000		X10,"QUE EL DEFINA, EL CONTENIDO DEL ARREGLO"/,	00001000
1100		X10,"AAAII I=1,2,...,TAM, QUE ES EL ARREGLO"/,	00001100
1200		X10,"DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS"/,	00001200
1300		X10,"VALORES DE LA SERIE QUE ESTA ALMACENADA EN"/,	00001300
1400		X10,"DISCO. TAM SE REFIERE AL TAMAÑO DE LA SERIE"/,	00001400
1410		X10,"POR EJEMPLO: SEAN ARRAY ARRI:1;REAL I"/,	00001410
1420		X10,"DECLARACIONES DE LA RUTINA, ENTONCES LAS"/,	00001420
1430		X10,"INSTRUCCIONES"/,	00001430
1440		X10," RESIZE(ARR,TAM);"/,	00001440
1450		X10," FOR I:=1 STEP 1 UNTIL TAM DO"/,	00001450
1460		X10," ARRI:=AAAII;"/,	00001460
1470		X10,"CUMPLIRIAN ESTE OBJETIVO."/,	00001470
1480		X10,"DESPUES DEBE CARGAR LOS NUEVOS VALORES EN"/,	00001480
1490		X10,"CCCCII PARA ESCRITURA EN DISCO. *">);	00001490
1500		END.	00001500

70	Z		00000070
80	Z	SECCION 2.3.4.18 COMANDO SS/2(FILTRA) ABREVIACION S12.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	[LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE FILTRA2;	00000500
600		BEGIN	00000600
700		WRITE(SAL,<"* OBSERVACION:"*,/,	00000700
710		X10,"ALGORITMO FILTRA2 NO IMPLEMENTADO",/,	00000710
800		X10,"EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO",/,	00000800
900		X10,"QUE EL DEFINA, EL CONTENIDO DEL ARREGLO",/,	00000900
1000		X10,"AAAI I I =1,2,...,TAM, QUE ES EL ARREGLO DONDE",/,	00001000
1100		X10,"SE CARGAN EN MEMORIA PRINCIPAL LOS VALORES DE",/,	00001100
1200		X10,"LA SERIE QUE ESTA ALMACENADA EN DISCO."*>);	00001200
1300		END.	00001300

70	Z		00000070
80	Z	SECCION 2.3.4.19 COMANDO SS/3(FILTRA) ADEVIACION S13.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AJT03IND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	L		00000200
300	L	LOADINFU "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE FILTR3;	00000500
600		BEGIN	00000600
700		WRITECSAL,<"* DESERVACION:"//,	00000700
800		X10,"ALGORITMO FILTR3 NO IMPLEMENTADO",//	00000800
900		X10,"EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO",//	00000900
1000		X10,"QUE EL DEFINA, EL CONTENIDO DEL ARREGLO",//	00001000
1100		X10,"AAACIT I=1,2,...TAM, QUE ES EL ARREGLO",//	00001100
1200		X10,"DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS",//	00001200
1300		X10,"VALORES DE LA SERIE QUE ESTA ALMACENADA EN",//	00001300
1400		X10,"CISCC. *">);	00001400
1500		END.	00001500

70	Z		0000070
80	Z	SECCION 2.3.4.20 COMANDO SS/1(ESPECTRO) ABREVIACION S21.	0000080
90	Z		0000090
100	S	SET LEVEL 5	0000100
110	S	SET AUTOBIND	0000110
120	S	HST IS OBJECT/SIMASET	0000120
200	[0000200
300	S	LOADINFO =ISAL/SIMASET/INFO	0000300
400]		0000400
500		PROCEDURE ESPECTRO1;	0000500
600		BEGIN	0000600
700		WRITE(SAL, <<" OBSERVACION :"/,	0000700
800		X10, "ALGORITMO ESPECTRO1 NO IMPLEMENTADO"/,	0000800
900		X10, "EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	0000900
1000		X10, "QUE EL DEFINA, EL CONTENIDO DEL ARREGLO"/,	0001000
1100		X10, "AAAI I=1,2,...,TAM, QUE ES EL ARREGLO"/,	0001100
1200		X10, "DONDE SE CARGAN EN MEMORIA PRINCIPAL, LOS"/,	0001200
1300		X10, "VALORES DE LA SERIE QUE ESTA ALMACENADA EN"/,	0001300
1400		X10, "TISCC. *">);	0001400
1500		END.	0001500

70	Z		0000070
80	Z	SECCION 2.3.4.22 COMANDO SS/3(ESPECTRO) ABREVIACION S23.	0000080
90	Z		0000090
100	S	SET LEVEL 3	0000100
110	S	SET AUTOBIND	0000110
120	S	HOST IS OBJECT/SIMASET	0000120
200	[0000200
300	[LOADINFO "ISAL/SIMASET/INFO"	0000300
400]		0000400
500		PROCEDURE ESPECTRO3;	0000500
600		BEGIN	0000600
700		WRITECSAL, <<"* OBSERVACION :"/,	0000700
800		X10, "ALGORITMO ESPECTRO3 NO IMPLEMENTADO."/,	0000800
900		X10, "EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	0000900
1000		X10, "QUE EL DEFINA, EL CONTENIDO DEL ARREGLO"/,	0001000
1100		X10, "AAAF: I = 1, 2, ..., TAM, QUE ES EL ARREGLO"/,	0001100
1200		X10, "DONDE SE CARGAN EN MEMORIA PRINCIPAL LOS"/,	0001200
1300		X10, "VALORES DE LA SERIE QUE ESTA ALMACENADA EN"/,	0001300
1400		X10, "DISCC. *")");	0001400
1500		END.	0001500

70	Z		00000070
80	Z	SECCION 2.3.4.23 COMANDO SS/1(SUAVIZA) ABEVIACION S31.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	I		00000200
300	I	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE SUAVIZA1;	00000500
600		BEGIN	00000600
700		WRITE(SAL, <<"* OBSERVACION :"/,	00000700
800		X10, "ALGORITMO SUAVIZA1 NO IMPLEMENTADO."/,	00000800
900		X10, "EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	00000900
1000		X10, "QUE EL DEFINA, EL CONTENIDO DEL ARREGLO"/,	00001000
1100		X10, "AAACII I=1,2,...,IA", QUE ES EL ARREGLO DONDE"/,	00001100
1200		X10, "SE CARGAN EN MEMORIA PRINCIPAL LOS VALORES"/,	00001200
1300		X10, "DE LA SERIE QUE ESTA ALMACENADA EN DISCO. *">);	00001300
1400		END.	00001400

70	Z		00000070
80	Z	SECCION 2.3.4.24 COMANDO SS/2(SUAVIZA) ABREVIACION S32.	00000080
90	Z		00000090
100	\$	SET LEVEL 3	00000100
110	\$	SET AUTOBIND	00000110
120	\$	HOST IS OBJECT/SIMASET.	00000120
200	\$		00000200
300	\$	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE SUAVIZA2;	00000500
600		BEGIN	00000600
700		WRITE(SAL, <<"* OBSERVACION :"/,	00000700
800		X10, "ALGORITMO SUAVIZA2 NO IMPLEMENTADO."/,	00000800
900		X10, "EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	00000900
1000		X10, "QUE EL DEFINA EL CONTENIDO DEL ARREGLO"/,	00001000
1100		X10, "AAAI] I=1,2,..., TAM, QUE ES EL ARREGLO"/,	00001100
1200		X10, "DONDE SE CARGAN EN MEMORIA PRINCIPAL LOS"/,	00001200
1300		X10, "VALORES DE LA SERIE QUE ESTA ALMACENADA"/,	00001300
1400		X10, "FN CISCO . *">);	00001400
1500		END.	00001500

70	Z		0000070
80	Z	SECCION 2.3.4.25 COMANDO SS/3(SUAVIZA) ABBREVIACION S33.	0000080
90	Z		0000090
100	S	SET LEVEL 3	0000100
110	S	SET AUTOBIND	0000110
120	S	HOST IS OBJECT/SIMASET	0000120
200	[0000200
300	[LOADINFO "ISAL/SIMASET/INFO"	0000300
400]		0000400
500		PROCEDURE SUAVIZA3;	0000500
600		BEGIN	0000600
700		WRITECSAL,<"* OBSERVACION :"/,	0000700
800		X10,"ALGORITMO SUAVIZA3 NO IMPLEMENTADO."/,	0000800
900		X10,"EL PROGRAMADOR DEBE CARGAR EN UN ARREGLO"/,	0000900
1000		X10,"QUE EL DEFINA EL CONTENIDO DEL ARREGLO"/,	0001000
1100		X10,"AAACII I=1,2,...,TAM, QUE ES EL ARREGLO"/,	0001100
1200		X10,"DONDE SE CARGAN EN MEMORIA PRINCIPAL LOS"/,	0001200
1300		X10,"VALORES DE LA SERIE ALMACENADA EN DISCO. *">);	0001300
1400		END.	0001400

```

70          Z          00000070
80          Z          00000080
90          Z          00000090
100         S          00000100
110         S          00000110
120         S          00000120
200         S          00000200
300         S          00000300
400         S          00000400
500         S          00000500
600         S          00000600
700         S          00000700
800         S          00000800
900         S          00000900
1000        S          00001000
1100        S          00001100
1200        S          00001200
1300        S          00001300
1400        S          00001400
1500        S          00001500
1600        S          00001600
1700        S          00001700
1800        S          00001800
1900        S          00001900
2000        S          00002000
2100        S          00002100
2300        S          00002300
2400        S          00002400
2500        S          00002500
2600        S          00002600
2700        S          00002700
2800        S          00002800
2900        S          00002900
3000        S          00003000
3100        S          00003100
3200        S          00003200
3300        S          00003300
3400        S          00003400
3500        S          00003500
3600        S          00003600
3700        S          00003700
3800        S          00003800

Z SECCION 2.3.4.26 COMANDO SS/(ESCALAR) ABBREVIACION SSE.
Z
Z
S SET LEVEL 3
S SET AUTOBINO
S HOST IS OBJECT/SIMASET
I
ILOADINFO "ISAL/SIMASET/INFO"
I
  PROCEDURE ESCALAR;
  BEGIN
    REAL C,D,I,NUMERO;
    ARRAY H(0:0);
    EBCDIC ARRAY FORMATO (0:10);
    LABEL LL;
    WRITE(SAL,<"* ESCRIBA EL ESCALAR POR EL QUE QUIERE MODIFICAR",/,
          "          A TODOS LOS ELEMENTOS DE LA SERIE. *>");
    READ(ENT,/,NUMERO);
    RESIZE(CCC,TAM);
    FOR I:=1 STEP 1 UNTIL TAM DO
      CCC(I):=AAA(I)*NUMERO;
    WRITE(SAL,<"* DESEA CONOCER SUS RESULTADOS *>");
    READ(ENT,<A2>,H(0));
    IF H(0)="SI"
      THEN BEGIN
        LL:
          WRITE(SAL,<"* ESPECIFIQUE EL INTERVALO *>");
          READ(ENT,/,C,D);
          IF C < 1 OR D > TAM
            THEN BEGIN
              WRITE(SAL,<"*** ERROR EN EL INTERVALO ***>");
              WRITE(SAL,<<X1>,"EL TAMANO DE LA SERIE VARIA DE 1 A",I6,>,TAM);
              GO TO LL;
            END
          ELSE
            FOR I:=C STEP 1 UNTIL D DO
              BEGIN
                REPLACE FORMATO BY " " FOR 1;
                REPLACE FORMATO BY "DATO ("&I&" FOR * DIGITS,"1";
                WRITE(SAL,<<X10,A11,F20.5>,FORMATO,CCC(I));
              END;
            END;
          END;
        END.

```



```

70      00000070
80      00000080
90      00000090
100     00000100
110     00000110
120     00000120
200     00000200
300     00000300
400     00000400
500     00000500
600     00000600
700     00000700
800     00000800
900     00000900
1000    00001000
1100    00001100
1200    00001200
1300    00001300
1400    00001400
1500    00001500
1600    00001600
1700    00001700
1800    00001800
1900    00001900
2000    00002000
2100    00002100
2200    00002200
2300    00002300
2400    00002400
2500    00002500
2600    00002600
2700    00002700
2800    00002800
2900    00002900
3000    00003000
3100    00003100
3200    00003200
3300    00003300
3400    00003400
3500    00003500
3600    00003600
3700    00003700
3800    00003800
3900    00003900
4000    00004000
4100    00004100
4200    00004200
4300    00004300
4400    00004400
4500    00004500
4600    00004600
4700    00004700
4800    00004800
4900    00004900
5000    00005000
5100    00005100
5200    00005200
5300    00005300
5400    00005400
5500    00005500
5600    00005600
5700    00005700
5800    00005800

Z SECCION 2.3.4 INCISO 5.
Z
S SET LEVEL 5
S SET AUTOBIND
S HOST IS OBJECT/SIMASET
[
S LOADINFO "ISAL/SIMASET/INFO"
PROCEDURE IDENTIFICA1;EXTERNAL;
PROCEDURE IDENTIFICA2;EXTERNAL;
PROCEDURE IDENTIFICA3;EXTERNAL;
PROCEDURE HISTOGRAMA1;EXTERNAL;
PROCEDURE HISTOGRAMA2;EXTERNAL;
PROCEDURE HISTOGRAMA3;EXTERNAL;
]
PROCEDURE SETINICIALIZA(F,G,S);
PROCEDURE INICIALIZA;
PROCEDURE F(X1,X2);
  FILE X1,X2;
  FORMAL;
PROCEDURE G(F,X1,X2);
  REAL G;
  FILE X1,X2;
  FORMAL;
REAL S;
BEGIN
  LABEL L2;
  ARPAY H0:=0;
  PROCEDURE MENSAJE(K);
    REAL K;
    BEGIN
      IF K=0 THEN
        WRITE(SAL,<<" SE EJECUTA EL COMANDO IDENTIFICA *>");
      IF K=1 THEN
        WRITE(SAL,<<" EL COMANDO OBTIENE EL HISTOGRAMA *>");
        WRITE(SAL,<<" EL COMANDO TRANSFORMA A LOS ELEMENTOS",/,
          " DE UNA SERIE EN ELEMENTOS DE UNA TABLA",/,
          " PARA LECTURA EJECUTA LEESERIE",/,
          " PARA ESCRITURA EJECUTA ESCRITABLA. *>");
      END;
  PROCEDURE CHEQUEA;
    IF FOCO=2
      THEN GJ L2;
  INICIALIZA;
  IF S=0 OR S=1 OR S=2
  THEN BEGIN
    MENSAJE(0);
    FOCO:=1;
    F(ARCH1,DIPECTORIO);
    CHEQUEA;
  END;
  IF S=3 OR S=4 OR S=5
  THEN BEGIN
    MENSAJE(1);
    FOCO:=1;
    F(ARCH1,DIPECTORIO);
    CHEQUEA;
  END;
CASE S OF
BEGIN
  IDENTIFICA1;
  IDENTIFICA2;
  IDENTIFICA3;

```


70	Z		00000070
80	Z	SECCION 2.3.4.27 COMANDO SET/IDENT) ABBREVIACION ST1.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	I		00000200
300	I	LOADINFO "ISAL/SIMASET/INFO"	00000300
400	I		00000400
500		PROCEDURE IDENTIFICAL;	00000500
600		BEGIN	00000600
700		WRITECSAL << "OBSERVACION :"/,	00000700
800		X10, "ALGORITMO IDENTIFICAL NO IMPLEMENTADO."/,	00000800
900		X10, "EL PROGRAMADOR DEBE DEFINIR EN ESTA RUTINA"/,	00000900
1000		X10, "UN ARREGLO EN EL QUE VA A TRANSFERIR LOS"/,	00001000
1100		X10, "DATOS DE LA SERIE QUE ESTAN EN UN ARREGLO"/,	00001100
1200		X10, "DE MEMORIA PRINCIPAL A AAIJ I=1,2,...,TAM"/,	00001200
1300		X10, "(TAM ES EL TAMANO DE LA SERIE), PARA REA"/,	00001300
1400		X10, "LIZAR LAS TRANSFORMACIONES PERTINENTES."/,	00001400
1500		X10, "SI LUEGO DESEA ALMACENAR EN DISCO LOS NUE"/,	00001500
1600		X10, "VOS VALORES OBTENIDOS, LOS DEBE CARGAR COMPLE"/,	00001600
1700		X10, "TAMENTE AL ARREGLO CCCII, CUYO TAMANO DE"/,	00001700
1800		X10, "BE ESTAR ACTUALIZADO, YA QUE LA RUTINA DE"/,	00001800
1900		X10, "ESCRITURA ESCRITABLE UTILIZA ESTE ARREGLO."/,	00001900
2000		X10, "POR OTRO LADO EL PROGRAMADOR DEBE PONER MU"/,	00002000
2100		X10, "CHA ATENCION EN LA ORGANIZACION DE LAS TA"/,	00002100
2200		X10, "BLAS (SOBRE TODO AL ORDEN N, FILAS M, COLUM P)"/,	00002200
2300		X10, "SI EL NUMERO DE DATOS QUE ESTAN EN CCCII"/,	00002300
2400		X10, "NO PONDERA CON EL PRODUCTO (N+1)*N*P SE"/,	00002400
2500		X10, "PRODUCIRAN ERRORES. **>);	00002500
2600		END.	00002600

70	Z		00000070
80	Z	SECCION 2.3.4.23 COMANDO SET/2(IDENT) ABREVIACION ST2.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300]	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE IDENTIFICA2	00000500
600		BEGIN	00000600
700		WRITE(SAL, <<"* OBSERVACION :"/,	00000700
800		X10,"ALCORRIMO IDENTIFICA2 NO IMPLEMENTADO."/,	00000800
900		X10,"EL PROGRAMADOR DEBE EJECUTAR EL COMANDO"/,	00000900
1000		X10,"SET/1(IDENT) PARA LAS ACCIONES GENERALES"/,	00001000
1100		X10,"PERTINENTES EN LA IMPLEMENTACION DE LA"/,	00001100
1200		X10,"ROUTINA. *"/>>);	00001200
1300		END.	00001300

70	Z		00000070
80	Z	SECCION 2.3.4.29 COMANDO SET/3(IDENT) ABREVIACION ST3.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300]	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE IDENTIFICA3;	00000500
600		BEGIN	00000600
700		WRITE(SAL,<<"* OBSERVACION :"/,	00000700
800		X10,"ALGORITMO IDENTIFICA3 NO IMPLEMENTADO."/,	00000800
900		X10,"EL PROGRAMADOR DEBE EJECUTAR EL COMANDO"/,	00000900
1000		X10,"SET/1(IDENT) PARA LAS ACCIONES GENERALES"/,	00001000
1100		X10,"PERTINENTES EN LA IMPLEMENTACION DE LA"/,	00001100
1200		X1,"ROUTINA. *"/);	00001200
1300		END.	00001300

70	Z		00000070
80	Z	SECCION 2.3.4.10 COMANDO SET/1(HISTO) ABBREVIACION ST4.	00000080
90	Z		00000090
100	I	SET LEVEL 3	00000100
110	I	SET AUTOBIND	00000110
120	I	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300	I	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE HISTOGRAMA1;	00000500
600		BEGIN	00000600
700		WRITE(SAL, <"* OBSERVACION :"/,	00000700
800		X10,"ALGORITMO HISTOGRAMA1 NO IMPLEMENTADO."/,	00000800
900		X10,"SI EL PROGRAMADOR DESEA MAYOR INFORMACION,"/,	00000900
1000		X10,"EJECUTE EL COMANDO SET/1(IDENT). *->");	00001000
1100		END.	00001100

70	Z	SECCION 2.3.4.31 COMANDE SET/2(HISTO) ABREVIACION ST5.	00000070
80	Z		00000080
90	S	SET LEVEL 3	00000090
100	S	SET AUTOBIND	00000100
110	S	HOST 'S OBJECT/SIMASET	00000110
120	L		00000120
200	S	LOADINFO "ISAL/SIMASET/INFO"	00000200
300	J		00000300
400		PROCEDURE HISTOGRAMA2;	00000400
500		BEGIN	00000500
600		WRITE(SAL.<"* OBSERVACION :",/,	00000600
700		X10,"ALGORITMO HISTOGRAMA2 NO IMPLEMENTADO.",/,	00000700
800		X10,"SI EL PROGRAMADOR DESEA MAYOR INFORMACION.",/,	00000800
900		X10,"EJECUTE EL COMANDO SET/1(IDENT). *">);	00000900
1000		END.	00001000
1100			00001100

70	Z		0000070
80	Z	SECCION 2.3.4.32 COMANDO SET/3(HISTO) ABBREVIACION ST6.	0000080
90	Z		0000090
100	S	SET LEVEL 3	0000100
110	S	SET AUTOBIND	0000110
120	S	HOST IS OBJECT/SIMASET	0000120
200	[0000200
300]	LOADINFO "ISAL/SIMASET/INFD"	0000300
400]		0000400
500		PROCEDURE HISTOGRAMA3;	0000500
600		BEGIN	0000600
700		WRITECSAL, <<"* OBSERVACION :"/,	0000700
800		X10, "ALGORITMO HISTOGRAMA3 NO IMPLEMENTADO."/,	0000800
900		X10, "SI EL PROGRAMADOR DESEA MAYOR INFORMACION,"/,	0000900
1000		X10, "EJECUTE EL COMANDO SET/1(IDENT). *">);	0001000
1100		END.	0001100


```

70      Z
80      Z SECCION 2.3.4 INCISO 7.
90      Z
100     S SET LEVEL 3
110     S SET AUTOBIAD
120     S HOST IS OBJECT/SIMASET
200     L
300     SLOADINFO "ISAL/SIMASET/INFO"
400     PROCEDURE SIMULA1;EXTERNAL;
500     PROCEDURE SIMULA2;EXTERNAL;
600     PROCEDURE SIMULA3;EXTERNAL;
700     I
800     PROCEDURE TAS(INICIALIZA,F,G,S);
900     PROCEDURE INICIALIZA;
1000    PROCEDURE F(X1,X2);
1100    FILE X1,X2;
1200    FORMAL;
1300    PROCEDURE G(G,X1,X2);
1400    REAL G;
1500    FILE X1,X2;
1600    FORMAL;
1700    REAL S;
1800    BEGIN
1900    LABEL L2;
2000    ARRAY HIO(0);
2100    PROCEDURE MENSAJE(K);
2200    REAL K;
2300    IF K=0 THEN
2400    WRITE(SAL,<<"* SE EJECUTA EL COMANDO SIMULA *",/);
2500    ** EL COMANDO SIMULA UNA SERIE A PARTIR",/;
2600    ** DE LOS ELEMENTOS DE UNA TABLA. *",/;
2700    ** PARA LECTURA EJECUTA EL COMANDO LEETABLA *",/;
2800    ** PARA ESCRITURA EJECUTA ESCRSERIE. *">>);
2900    PROCEDURE CHEQUEA;
3000    IF FOCO=2
3100    THEN GO L2;
3200    INICIALIZA;
3300    IF S=0 OR S=1 OR S=2
3400    THEN BEGIN
3500    MENSAJE(0);
3600    FOCO:=1;
3700    FCTABLA,DIRTAB);
3800    CHEQUEA;
3900    END;
4000    CASE S OF
4100    BEGIN
4200    SIMULA1;
4300    SIMULA2;
4400    SIMULA3;
4500    END;
4600    WRITE(SAL,<<"* DESEA ALMACENAR LOS VALORES OBTENIDOS *">>);
4700    READ(ENT,<<A2>>,H(1));
4800    IF H(1)="SI"
4900    THEN BEGIN
5000    FOCO:=1;
5100    G(G,ARCH1,DIRECTORIO);
5200    END;
5300    WRITE(SAL,<<"RR",/);
5400    L2:
5500    END.

```

70	Z		00000070
80	Z	SECCION 2.3.4.33 COMANDO TAS/1(SIMULA) ABBREVIACION TS1.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOLOAD	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300]	LOADINFO "ISAL/SIMASET/INFO"	00000300
400]		00000400
500		PROCEDURE SIMULA1;	00000500
600		BEGIN	00000600
700		WRITECSAL <<"* OBSERVACION :"/,	00000700
800		X10,"ALGORITMO SIMULA1 NO IMPLEMENTADO."/,	00000800
900		X10,"LAS INDICACIONES GENERALES PARA IMPLEMENTAR"/,	00000900
1000		X10,"ESTA RUTINA SIN LAS MISMAS QUE SE OBTIENEN"/,	00001000
1100		X10,"AL EJECUTAR EL COMANDO SET/1(IDENT) , CON"/,	00001100
1200		X10,"LA DIFERENCIA QUE AL MOMENTO DE ESCRITURA"/,	00001200
1300		X10,"EN DISCO SE EJECUTA EL COMANDO ESCRSERIE."/,	00001300
1400		X10,"ESTO YA ESTA PREVISTO POR EL SISTEMA. *">);	00001400
1500		END.	00001500

70	2	SECCION 2.3.4.34 COMANDO TAS/2(SIMULA) ABEVIACION TS2.	00000670
80			00000680
90	3	SET LEVEL 3	00000690
100			00000700
110	3	SET AUTOBIND	00000710
120	3	HOST IS OBJECT/SIMASET	00000720
130			00000730
140	[LOADINFO "ISAL/SIMASET/INFO"	00000740
150			00000750
160]		00000760
170		PROCEDURE SIMULA2;	00000770
180		BEGIN	00000780
190		WRITECSAL, <"* OBSERVACION :"/,	00000790
200		X10, "ALGORITMO SIMULA2 NO IMPLEMENTADO."/,	00000800
210		X10, "LAS INDICACIONES GENERALES PARA IMPLEMENTAR"/,	00000810
220		X10, "ESTA RUTINA SON LAS MISMAS QUE SE OBTIENEN"/,	00000820
230		X10, "AL EJECUTAR EL COMANDO SET/ICIDENT) , CON LA"/,	00000830
240		X10, "DIFERENCIA QUE EL SISTEMA PARA ALMACENAR EN"/,	00000840
250		X10, "DISCO LOS NUEVOS RESULTADOS ACTIVA LA RUTINA"/,	00000850
260		X10, "ESCRIBIR. **>);	00000860
270		END.	00000870

70	Z		00000070
80	Z	SECCION 2.3.4.35 COMANDO TAS/3(SIMULA) ABREVIACION TS3.	00000080
90	Z		00000090
100	S	SET LEVEL 3	00000100
110	S	SET AUTOBIND	00000110
120	S	HOST IS OBJECT/SIMASET	00000120
200	[00000200
300		\$LOADINFO "ISAL/SIMASET/INF"	00000300
400]		00000400
500		PROCEDURE SIMULA3;	00000500
600		BEGIN	00000600
700		WRITE(SAL, <<"* OBSERVACION :"/,	00000700
800		X10, "ALGORITMO SIMULA3 NO IMPLEMENTADO."/,	00000800
900		X10, "LAS INDICACIONES GENERALES PARA IMPLEMENTAR"/,	00000900
1000		X10, "ESTA ROTINA, SON LAS MISMAS QUE SE OBTIENEN"/,	00001000
1100		X10, "AL EJECUTAR EL COMANDO SET/1(IDENT), CON LA"/,	00001100
1200		X10, "DIFERENCIA DE QUE EL SISTEMA ACTIVA LA ROTINA"/,	00001200
1300		X10, "ESCRIBIR, PARA EL CASO DE ALMACENAR EN DISCO"/,	00001300
1400		X10, "LOS NUEVOS VALORES OBTENIDOS. *">);	00001400
1500		END.	00001500

```

70      Z
80      Z SECCION 2.3.4 INCISO 3.
90      Z
100     $ SET LEVEL 3
110     $ SET AUTOBIND
120     $ HOST IS OBJECT/SIMASET
130     $ BIND GRAFIK FROM (ISFB)ORCL/HP/GRAF
200     E
300     $LOADINFO "YSAL/SIMASET/INFO"
400     PROCEDURE GRAFIK(R,A,B,C,D);
500     VALUE R,A,B,C,D; REAL R,A,B,C,D;
600     EXTERNAL;
700
800     1
900     PROCEDURE GRACINICIALIZA(F,C);
1000    PROCEDURE INICIALIZA;
1100    PROCEDURE F(X1,X2);
1200    FILE X1,X2;
1300    FORMAL;
1400    REAL G;
1500    BEGIN
1600    LABEL L2;
1700    REAL I,J,J1,J2,MAX1,MIN1,X,Y,L;
1800    PROCEDURE MFNSAJE(K);
1900    REAL X;
2000    BEGIN
2100    WRITE(SAL,<"* ALGORITMO DE GRAFICACION *">);
2200    IF K=0 THEN
2300    WRITE(SAL,<"* OBSERVACION :"/,
2400    X7,"EL ALGORITMO UNE LOS DIVERSOS PUNTOS EN"/,
2500    X7,"FORMA CONTINUA, ES DECIR SE UNEN LOS PUNTOS"/,
2600    X7,"P1(X1,Y1),P2(X2,Y2),...,PN(XN,YN). *">);
2700    IF K=1 THEN
2800    WRITE(SAL,<"* OBSERVACION :"/,
2900    X7,"EL ALGORITMO UNE LOS PUNTOS"/,
3000    X7,"P1(X1,Y1),P2(X2,Y1),P3(X2,Y2),P4(X3,Y2),..."/,
3100    X7,"EN FORMA DE HISTOGRAMA. *">);
3200    IF K=2 THEN
3300    WRITE(SAL,<"* OBSERVACION :"/,
3400    X7,"EL ALGORITMO UNE LOS PUNTOS"/,
3500    X7,"P1(X1,Y1),P2(X1,Y2),P3(X2,Y2),P4(X2,Y3),..."/,
3600    X7,"EN FORMA DE HISTOGRAMA. *">);
3700    IF K=3 THEN
3800    WRITE(SAL,<"* OBSERVACION :"/,
3900    X7,"EL ALGORITMO DIBUJA LINEAS QUE"/,
4000    X7,"PARTIENDO DE LOS PUNTOS P1*(X1,0),P2*(X2,0),..."/,
4100    X7,"TERMINAN EN LOS PUNTOS P1(X1,Y1),P2(X2,Y2),..."/,
4200    X7,"RESPECTIVAMENTE . *">);
4300    IF K=4 THEN
4400    WRITE(SAL,<"* OBSERVACION :"/,
4500    X7,"EL ALGORITMO USICA LOS PUNTOS"/,
4600    X7,"P1(X1,Y1),P2(X2,Y2),...,PN(XN,YN)"/,
4700    X7,"EN EL PLANO FORMADO POR LAS COORDENADAS X E Y. *">);
4800    WRITE(SAL,<"* EJECUTA PRIMERO EL COMANDO LEESERIE *">);
4900    END;
5000    PROCEDURE CHEQUEA;
5100    IF FOC=2
5200    THEN GO L2;
5300    PROCEDURE OPTIENE;
5400    BEGIN
5500    MINI:=MAX1:=AAAL1;
5600    J1:=J2:=1;
5700    FOR I=1 STEP 1 UNTIL JAM DO

```

```

5800           IF AAA[I] > MAX1
5900           THEN BEGIN
6000             MAX1:=AAA[I];
6100             J1:=I;
6200           END;
6300           IF AAA[I] < MIN1
6400           THEN BEGIN
6500             MIN1:=AAA[I];
6600             J2:=I;
6700           END;
6800         END;
6900       END;
7000 Z
7100 Z X SECCION 2.3.4.36 COMANDO GRA/1(CONT) ABREVIACION GR1.
7200 Z
7300       PROCEDURE GRAFICO1;
7400       BEGIN
7500         GRAFIK("DEFR",-1,MIN1-2,TAM+1,MAX1+2);
7600         GRAFIK("EJES",0,0,0,0);
7700         X:=1;
7800         Y:=AAA[I];
7900         GRAFIK("MOVR",X,Y,0,0);
8000         GRAFIK("BAJA",0,0,0,0);
8100         FOR X:=1 STEP 1 UNTIL TAM DO
8200           GRAFIK("MOVR",X,AAA[X],0,0);
8300           GRAFIK("LEVA",X,0,0,0);
8400         END;
8500 Z
8600 Z X SECCION 2.3.4.37 COMANDO GRA/2(HISTO) ABREVIACION GR2.
8700 Z
8800       PROCEDURE GRAFICO2;
8900       BEGIN
9000         GRAFIK("DEFR",-1,MIN1-2,TAM+1,MAX1+2);
9100         GRAFIK("EJES",0,0,0,0);
9200         X:=1;
9300         Y:=AAA[I];
9400         GRAFIK("MOVR",X,Y,0,0);
9500         GRAFIK("BAJA",0,0,0,0);
9600         FOR X:=1 STEP 1 UNTIL TAM DO
9700           BEGIN
9800             GRAFIK("MOVR",X,AAA[X],0,0);
9900             GRAFIK("MOVR",X+1,AAA[X],0,0);
10000          END;
10100         GRAFIK("LEVA",0,0,0,0);
10200       END;
10300 Z
10400 Z X SECCION 2.3.4.38 COMANDO GRA/3(HISTO) ABREVIACION GR3.
10500 Z
10600       PROCEDURE GRAFICO3;
10700       BEGIN
10800         GRAFIK("DEFR",-1,MIN1-2,TAM+1,MAX1+2);
10900         GRAFIK("EJES",0,0,0,0);
11000         X:=1;
11100         Y:=AAA[I];
11200         GRAFIK("MOVR",X,Y,0,0);
11300         GRAFIK("BAJA",0,0,0,0);
11400         FOR X:=1 STEP 1 UNTIL TAM-1 DO
11500           BEGIN
11600             GRAFIK("MOVR",X,AAA[X],0,0);
11700             GRAFIK("MOVR",X,AAA[X+1],0,0);
11800           END;
11900         GRAFIK("MOVR",TAM,AAA[TAM],0,0);
12000         GRAFIK("LEVA",0,0,0,0);
12100       END;
12200 Z
12300 Z X SECCION 2.3.4.39 COMANDO GRA/4(LINEAS) ABREVIACION GRA

```

```

00005800
00005900
00006000
00006100
00006200
00006300
00006400
00006500
00006600
00006700
00006800
00006900
00006910
00006920
00006930
00007000
00007100
00007200
00007300
00007400
00007500
00007600
00007700
00007800
00007900
00008000
00008100
00008110
00008120
00008130
00008200
00008300
00008400
00008500
00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009610
00009620
00009630
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00010910
00011000
00011100
00011110
00011120

```

11130	Z	PROCEDURE GRAFICO4;	00011130
11200		BEGIN	000111200
11300		GRAFIK("DEFR",-1,MIN1-2,TAM+1,MAX1+2);	000111300
11400		GRAFIK("EJES",X,0,0,0);	00011400
11500		X:=1;	00011500
11600		Y:=AAA[X];	00011600
11700		GRAFIK("MOVR",X,Y,0,0);	00011700
11800		GRAFIK("BAJA",X,0,0,0);	00011800
11900		FOR X:=1 STEP 1 UNTIL TAM DO	00011900
12000		GRAFIK("LINR",X,0,X,AAA[X]);	00012000
12100		END;	00012100
12200			00012200
12210	Z		00012210
12220	Z	SECCION 2.3.4.40 COMANDO GRA/S(PUNTOS) ABBREVIACION GRS.	00012220
12230	Z		00012230
12300		PROCEDURE GRAFICO5;	00012300
12400		BEGIN	00012400
12500		GRAFIK("DEFR",-1,MIN1-2,TAM+1,MAX1+2);	00012500
12600		GRAFIK("EJES",X,0,0,0);	00012600
12700		FOR X:=1 STEP 1 UNTIL TAM DO	00012700
12900		GRAFIK("PUNR",X,AAA[X],0,0);	00012900
13100		END;	00013100
13200		INICIALIZA;	00013200
13300		CASE 0 OF	00013300
13400		BEGIN	00013400
13500		BEGIN	00013500
13600		MENSAJE(0);	00013600
13700		FOCC:=1;	00013700
13800		F(CARCHI,DIRECTORIO);	00013800
13900		CHECUFA;	00013900
14000		OBTIENE;	00014000
14100		GRAFICO1;	00014100
14200		END;	00014200
14300		BEGIN	00014300
14400		MENSAJE(1);	00014400
14500		FOCC:=1;	00014500
14600		F(CARCHI,DIRECTORIO);	00014600
14700		CHECUFA;	00014700
14800		OBTIENE;	00014800
14900		GRAFICO2;	00014900
15000		END;	00015000
15100		BEGIN	00015100
15200		MENSAJE(2);	00015200
15300		FOCC:=1;	00015300
15400		F(CARCHI,DIRECTORIO);	00015400
15500		CHECUFA;	00015500
15600		OBTIENE;	00015600
15700		GRAFICO3;	00015700
15800		END;	00015800
15900		BEGIN	00015900
16000		MENSAJE(3);	00016000
16100		FOCC:=1;	00016100
16200		F(CARCHI,DIRECTORIO);	00016200
16300		CHECUFA;	00016300
16400		OBTIENE;	00016400
16500		GRAFICO4;	00016500
16600		END;	00016600
16700		BEGIN	00016700
16800		MENSAJE(4);	00016800
16900		FOCC:=1;	00016900
17000		F(CARCHI,DIRECTORIO);	00017000
17100		CHECUFA;	00017100
17200		OBTIENE;	00017200
17300		GRAFICO5;	00017300
17400		END;	00017400
17500		END;	00017500

17600	IF J1 > J2	00017600
17700	THEN L:=J2/TAM	00017700
17800	ELSE L:=J1/TAM	00017800
17900	FOR X:=1 STEP 1 UNTIL TAM DO	00017900
18000	GRAFIK("LINF",K,MAX1/100,X,0);	00018000
18100	GRAFIK("LTNR",J1,(2+MAX1),J1,(1+MAX1));	00018100
18200	GRAFIK("LTNR",J1-L,MAX1+2,J1+L,MAX1+2);	00018200
18300	GRAFIK("LTNR",J2-MIN1-1,J2-MIN1-2);	00018300
18400	GRAFIK("LTNR",J2-L,MIN1-2,J2+L,MIN1-2);	00018400
18500	GRAFIK("APAG",0,0,0,0);	00018500
18600	WRITE(SAL,<"NN",?>);	00018600
18700	L2:	00018700
18800	END.	00018800